PCT WORLD INTELLECTUAL PRO
International I

INTERNATIONAL APPLICATION PUBLISHED UNDE

WO 9606403A1

(51) International Patent Classification 6 :
G06F 17/60, G07F 17/26

A1

(11) I

(43) International Publication Date: 29 February 1996 (29.02.96)

(21) International Application Number: PCT/US95/10518

(22) International Filing Date: 18 August 1995 (18.08.95)

(30) Priority Data:
08/292,611 18 August 1994 (18.08.94) US
08/472,898 7 June 1995 (07.06.95) US

(71) Applicant: CREATACARD, INC. [US/US]; One American Road, Cleveland, OH 44114 (US).

(72) Inventors: ALLRED, Scott, K.; 3225 Sabrina Court, Las Vegas, NV 89117 (US). HELTON, Mike, D.; 3360 Paso Andres, Las Vegas, NV 89102 (US). RUSSELL, H., Matthew; 32 Princeville Lane, Las Vegas, NV 89113 (US). STOKES, William, S.; 2101 San Miguel, North Las Vegas, NV 89030 (US).

(74) Agents: COCHRAN, David, B. et al.; Jones, Day, Reavis & Pogue, North Point, 901 Lakeside Avenue, Cleveland, OH 44114 (US).

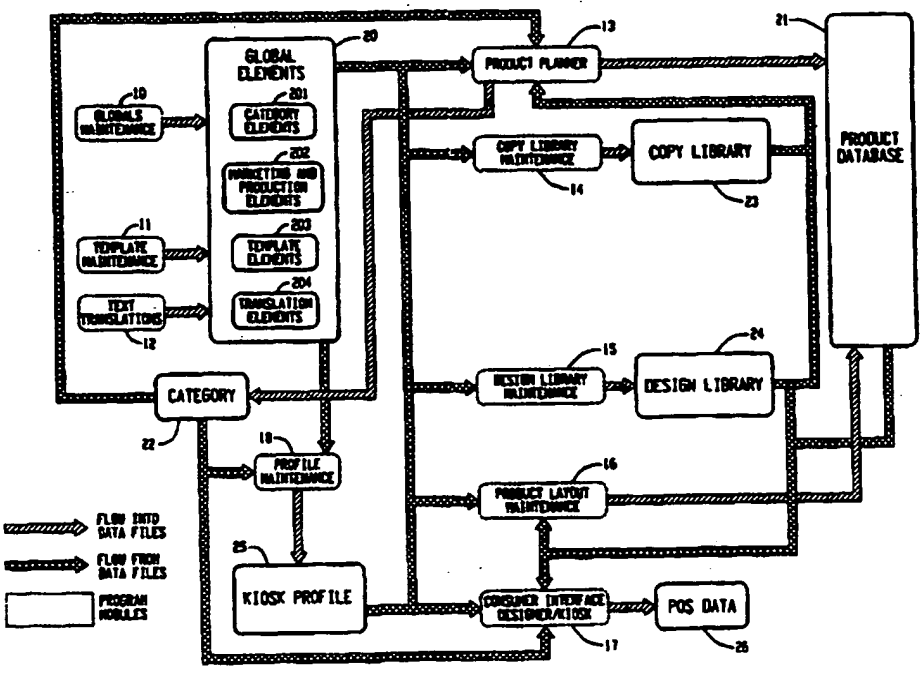
(81) Designated States: AU, CA, GB, MX, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR THE DEVELOPMENT AND IMPLEMENTATION OF AN INTERACTIVE AND DYNAMICALLY RESPONSIVE CUSTOMER SERVICE SYSTEM

(57) Abstract

A customer service system interface development tool is disclosed for use by an interface designer in creating an interface for customer interaction. The interface created by the development tool may be incorporated into a customer service system for presenting products or services to a customer for the customer to make a product or service selection if the customer so desires from the products or services presented as a result of the customer's interaction with the interface. The interface tool includes modules for specifying global parameters relating products or services to be presented to the customer through the interface and developing a profile of the customer service system environment in which the interface is to operate. An additional module aids the interface designer in planning a presentation by associating a set of presentation data with the products or services available for presentation to the customer. Optional modules include modules for planning products for production at the same location as the customer service system embodying the interface. An example of use of the interface development tool in the greeting card environment is also disclosed.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**GB**	United Kingdom	**MR**	Mauritania
AU	Australia	**GE**	Georgia	**MW**	Malawi
BB	Barbados	**GN**	Guinea	**NE**	Niger
BE	Belgium	**GR**	Greece	**NL**	Netherlands
BF	Burkina Faso	**HU**	Hungary	**NO**	Norway
BG	Bulgaria	**IE**	Ireland	**NZ**	New Zealand
BJ	Benin	**IT**	Italy	**PL**	Poland
BR	Brazil	**JP**	Japan	**PT**	Portugal
BY	Belarus	**KE**	Kenya	**RO**	Romania
CA	Canada	**KG**	Kyrgystan	**RU**	Russian Federation
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CG	Congo			**SE**	Sweden
CH	Switzerland	**KR**	Republic of Korea	**SI**	Slovenia
CI	Côte d'Ivoire	**KZ**	Kazakhstan	**SK**	Slovakia
CM	Cameroon	**LI**	Liechtenstein	**SN**	Senegal
CN	China	**LK**	Sri Lanka	**TD**	Chad
CS	Czechoslovakia	**LU**	Luxembourg	**TG**	Togo
CZ	Czech Republic	**LV**	Latvia	**TJ**	Tajikistan
DE	Germany	**MC**	Monaco	**TT**	Trinidad and Tobago
DK	Denmark	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**US**	United States of America
FI	Finland	**ML**	Mali	**UZ**	Uzbekistan
FR	France	**MN**	Mongolia	**VN**	Viet Nam
GA	Gabon				

Method and apparatus for the development and implementation of an interactive and dynamically responsive customer service system.

FIELD OF INVENTION

The invention relates to a method and apparatus for developing interactive customer service systems for the sale or display of products or services, wherein each such system is dynamically responsive to changed marketing conditions and consumer-indicated presentation preferences and each system is reconfigurable in light of the changed marketing conditions to present a selection of products or services that corresponds to the demographic and other marketing information developed through on-site use of the system. Also disclosed herein is an interactive customer service system which incorporates a dynamically alterable customer interface designed using the method and apparatus of the present invention.

BACKGROUND OF THE INVENTION

During the late 1980's and early 1990's, several computer-controlled vending systems were developed to enable the vending of greeting cards and related products. One such system that employs computer control of the vending process is represented by U.S. Patent No. 5,056,472 issued to Buckley et al. and assigned to Hallmark Cards, Inc. The '472 patent describes a greeting card vending machine where stacks of different partially-printed cards are retrieved, personalized, and then dispensed to the customer. A computer keyboard is presented to permit the customer to select from among the available pre-printed card stock and to insert personal

messages or information to personalize the card selected. In the embodiment set forth in the '472 patent, a robot-like structure, in conjunction with computer control, operates to deliver the selected pre-printed card from a physical storage area to a printer for the insertion of the personalized text and then to a delivery slot for retrieval by the customer. In one known commercial application of a machine similar to that disclosed in the Buckley '472 patent, Hallmark has provided for the selection of a pre-printed card directly by the customer from a standard greeting card display without the assistance of a robot-like mechanism. The customer delivers the card to a salesperson for insertion into a printer for the personalization in accordance with data previously presented by the customer through a keyboard or similar data entry device.

A different approach to computer-controlled vending of greeting cards and related products can be found in U.S. Patent No. 5,056,029, issued to Cannon. The Cannon patent discloses a system in which greeting card and other related product designs are stored in digital form in a data base that is searchable by a customer through a touch-screen or other data entry device. The Cannon system guides the customer's selection of a graphic design to be incorporated into a product through a series of successive menus that result in the selection of parameters that describe the occasion or purpose for which the product is to be used, or that describe the age and gender characteristics of either the recipient of the product or the sender of the product. Based upon the customer's selection of these parameters, the system locates a selection of products that corresponds to the selected parameters and presents the selection on a CRT display for the customer to peruse and to select one product design for production. The final product is then printed on a plotter or a printer that is co-located with the CRT display and touch-screen through which the

customer made his or her design selection. An additional feature of the Cannon system is an ability to transfer sales data from the vending location to a remote location.

These prior art systems along with other interactive point-of-sale and point-of-preview systems, although providing partial solutions to the need to automate the vending and production of products that are designed to meet consumer preferences, do not provide a method or apparatus for ensuring that changes in marketing strategies are reflected in the selections made available to the customers who use the systems (throughout this application the term customer and consumer are used interchangeably to refer to persons who preview and purchase products from a customer service system). Nor do the prior art systems provide the capability to assist those involved in product development and marketing to assign certain properties to products to be marketed so that an efficient, automated search strategy can be employed through an interface to the consumer that takes into consideration those properties.

In addition to the above listed shortcomings of prior art point-of-sale and point-of-preview systems, it is believed that no system has been developed to provide a system developer with a set of tools to produce an integrated customer service system with an interface that easily and dynamically is reconfigured to reflect changed marketing conditions or a change in the products or services to be offered through the customer service system without having to reprogram the software driving the interface. More specifically, it is believed that no presently available system exists that permits a kiosk interface designer or other interactive service provider to build and use a standardized, menu-driven interface to a customer service system, wherein the interface dynamically changes in response to a recognition of the time of year in which the interface is to be operated, or

in response to the setting of parameters indicating the country or language in which the interface is to be operated, or in response to accumulated statistics on consumer preferences for particular services or products.

Accordingly, it is an object of the present invention to provide a method and apparatus for the development and implementation of an automatic system for the vending of products, wherein the automatic system is responsive to certain inputs which enable the system to change dynamically the products offered by the system so that the preferences of a particular consumer with respect to the products marketed are met.

It is a further object of the invention to provide a system that manages the design, marketing and sale of certain products from the point at which the products are first conceived through the point that the products are selected by the consumer.

Still another object of the invention is to provide a dynamic user interface for a system for on-site production and vending of products wherein the user interface dynamically alters itself by taking into consideration the languages spoken in the region served by the system, other demographic traits associated with the likely users of the system, or the time of year in which the interface is operating.

Another object of the invention is to provide a management system that enables the production of a vending system that can be easily modified to accommodate new products, new market segments, new languages, and new methods of on-site production of such products.

A further object of the invention is to provide a vending system that presents a consumer with an interface that includes visual and nonvisual controls and a keyboard on a touch-screen and that permits the consumer to choose, through "on the fly" translation, the language displayed on the visual controls as well as the keyboard layout associated with the displayed language.

A further object of the invention is to provide a customer service system for permitting a customer to preview or purchase a product or service, wherein the customer service system includes memory, a processor, an input device, and a display for presenting to the customer the dynamically alterable customer interface designed using the method and apparatus disclosed herein.

Additional objects, advantages and novel features of the invention will be set forth in the description that follows. Other objects, advantages and novel features will become apparent to those skilled in the art who examine the description of the invention or through the practice of the invention as set forth below.

SUMMARY OF THE INVENTION

In achieving these and other objects, a method and apparatus has been provided for developing customer service systems for the marketing of products, wherein each such system is adaptable to the marketing environment in which the customer service system is to be deployed or to reflect a particular marketing emphasis within that environment, and further wherein each such customer service system is dynamically responsive to changes in that marketing environment. The method and apparatus according to the invention describe essentially a "metasystem" -- a system for producing or "authoring" other systems, particularly multimedia customer service systems for the marketing of products and services that are to be selected in light of certain preferences or product characteristics specified by a consumer. Such a customer service system may also permit a consumer to specify modifications or customizations to the product he or she selects through the system, and, if appropriate, the system may also play a role in managing the production of the selected product, either on-site or at a remote production location.

Several software modules in combination with several types of hardware suites or platforms perform various functions within the metasystem. For ease of description, but without limiting the generality of the invention, the metasystem shall be set forth in reference to the production of customer service systems or kiosks for the on-site production and sales of greeting cards, invitations, banners, posters, announcements, wrapping paper and like products. It should be appreciated by one of ordinary skill in the art that the metasystem described in greater detail below has application in designing any system for interaction with consumers of services, products or information in which a consumer is asked or directed through a series of menus to select, preview and/or purchase a product or service that meets the consumer's specifications, or simply to provide or receive information, such as in a customer preference survey system or an information directory kiosk. It should further be appreciated by one of ordinary skill in the art that a customer service system produced by the metasystem need not reside on or be provided through a customer services terminal or kiosk, but that such systems may operate through an interactive cable television system, an on-line information system such as CompuServe© or Prodigy©, or through an enhanced telecommunications service. It should also be appreciated that although the preferred embodiment contemplates that the customer service system produced by the metasystem will fulfill the customer's service or product purchase request proximate to the terminal through which the customer made his or her preferences known, remote fulfillment of the customer's request can also be accomplished through such a customer service system in accordance with the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing summary as well as the detailed description of the invention that follows are presented

by way of example and are not intended to limit the present invention solely to the embodiments described therein. Other objects, features and aspects of the invention will be best appreciated by one of ordinary skill in the art when the description of the invention is viewed in conjunction with the accompanying drawings briefly described as follows:

FIG. 1 is a data flow diagram of one of the preferred embodiments of the invention.

FIGS. 2, 3, 4, 5, 6, 7 and 8 are representations of the various tables, databases and other data sources referenced in the flow diagram of FIG. 1.

FIGS. 9, 10, 11 and 12 represent the object oriented class hierarchy for an alternative embodiment of the invention.

FIGS. 13, 14, 15, 16, 17, 18, 19, 20, 21, and 22 comprise a flow diagram for the main software module which enables the design, test, and execution of dynamic interfaces for use with a customer service system.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

In the preferred embodiment, the metasystem can be described logically as consisting of nine software modules and a number of associated data files. For ease of description, the modules shall be referred to, as in Figure 1, as the Globals Maintenance Module (10); Template Maintenance Module (11); Text Translations Module (12); Product Planner Module (13); Copy Library Maintenance Module (14); Design Library Maintenance Module (15); Product Layout Maintenance Module (16); Profile Maintenance Module (18) and Consumer Interface Designer/Kiosk Module (17). Each software module listed above resides, in the preferred embodiment, within the test and development unit and art work stations, while on field units where the actual vending of products or services may occur, only the Kiosk version of the Consumer Interface Designer/Kiosk Module (17) will

reside. The primary difference between the Kiosk or field version of the Consumer Interface Designer Module and the version which resides on the test and development unit workstations is that in the field version, a user cannot design and test custom interfaces, but is limited to the execution of a stored interface script (a script, as described below, is a sequence of interactive frames which define the interaction with the consumer). The Consumer Interface Designer Module (17), described more fully below, is the heart of the present invention. It permits an interface designer to design, test and execute the novel dynamically alterable customer interfaces of the present invention. The other 8 software modules provide for input, management, organization and storage of global data, product data, copy data, design data, and system profile data which the Interface Designer Module (17) utilizes in the creation of the custom interfaces.

While a variety of users may provide data for use with the metasystem, it is the interface designer, or "author" who conducts the primary interaction with the system in order to create or author the interface script which is displayed to a consumer interacting with a customer service system. Other users of the metasystem may include (1) a system administrator, who provides system management input and configuration control for the overall system, (2) a product designer, who inputs data relevant to the particular specifications of products or services that may be offered to consumers, (3) marketing personnel, who may input data related to the particular marketing conditions that an interface can be played in, (4) creative personnel, who may input data relevant to the artistic design of a particular product, (5) copy editors, who provide relevant text phases for use with a particular set of products, or (6) layout artists, who combine the artistic design and copy to form a final product for display to consumers. While it is envisoned that a variety of users within a product design

organization could interact with the metasystem, it is also possible that a single person, the interaface designer, could take responsibility for implementing all aspects of the design flow.

The customer service system produced by the metasystem according to the invention can be described from any point in the production process, but for ease of description will be described from the point a new product is conceived to the point a field unit embodying a customer service system is placed into operation. At the first stage of the invention, a new product line is conceived. For purposes of illustration, and without limiting the generality of the invention, the example of a system used to produce and market, through a consumer interface, a line of posters celebrating "European Unity Day" is used. During this stage, the Globals Maintenance Module is used to modify the Globals Elements (20) comprising the Category Elements (201) and the Marketing and Production Elements (202) to include information about the new product line that may be shared with other modules (see Figure 2). The Globals Maintenance Module (10) could be used in this example, through an associated dialog box to specify what countries will be available to the interface designer. The administrator of the metasystem might want to restrict, in this instance, the interface designer to interfaces only for all the European Union countries. The information supplied is stored in a Countries database associated with the invention under the table Countries in the manner shown in Table VI (Each Table included in this application has 5 columns; Column 1 indicates whether a variable is used by another table -- PK is a public key which indicates global access, FK is a foreign key, such variables are accessed by other defined data structures, and no symbol means the variable is local; Column 2 is the name of the variable; Column 3 is the variable type; Column 4 is the variable size; and Column 5 is a description of the

variable if necessary). The administrator can also choose to add the main languages spoken in the European Union countries to a Languages Database through a similar dialog box. Those language names are stored under the table Languages in the manner shown in Table IX. The administrator could also specify the type of keyboard layout associated with each language through the same dialog box.

To a ProductData Table (see Table XVIII), the administrator may add through a dialog box the product Posters and parameters defining what a poster product is. As can be seen in this table, a number of data items are associated with the product through dialog boxes, either globally or with respect to a particular instance of the metasystem's operation, such as the instructions for laying out the product, the countries in which the product is "visible," i.e., where it will be permitted to be marketed through the custom interface developed by the interface designer, the languages that the product will be visible in, etc. The name of the product is stored as well in the Products Table as set forth structurally on Table XIX.

The administrator can similarly add "European Unity Day" to the Occasions Table in the structure shown in Table XV and can add poster stock to the "PaperTypes" data base, the structure of which is set forth in Table XVI. Although in the preferred embodiment it is envisioned that data entry occurs through dialog boxes that are preprogrammed using similar coding techniques described more fully below as a part of the description of the Consumer Interface Designer/Kiosk Module (17), it would be well appreciated by one of ordinary skill in the art that any manner of placing data in a database can be used to establish the global parameters available to an interface designer.

In addition to each of these modifications, other items in the metasystem data base may similarly be

established globally. These include the types of printers available through the Printer Table (see Table XVII), licenses to characters appearing on products and designs through a Licenses Table (see Table IX), markets to which particular products are to be marketed through a Markets Table (see Table XIV). Other items which may be established or modified through the Globals Maintenance Module include the line list for list of products to be marketed (Table XI), text lists for use in displays and in all other areas in which the metasystem uses lists of several items (Table XII) and the names associated with such lists (see Table XIII).

After the Category Elements and the Marketing and Production Elements have been modified, at the second stage of the invention a new product template is created through the Template Maintenance Module (11). Through this module, a product designer can specify the paper size, the paper type, the printer selection, the paper orientation, a record of the dividing lines or folds associated with the particular product type for which a template is to be developed, as well as whether the product is to be printed in color or black and white. To continue with the example of a poster product, the paper size might be set as ''poster size'', the paper type might be set as "poster stock", the printer selection might be set as a plotter (such as a Hewlett-Packard HP 7550 plotter), the paper orientation might be set as "landscape", the color/black and white selector might be set to "color" and the record of dividing lines or folds might be set to "None". Thus, a new product template for a poster product line such as a "European Unity Day" poster product line is created through the Template Maintenance Module (11) and added to the Template Elements Database (203). The structure of the data records for the Product Template data is shown on Table XXIV.

It should be appreciated, however, that a product design may require more than one template. If, for example, both single sided and double sided posters are to be included in a product line, then one template may include a value indicating duplex printing, while another may include a value indicating one sided printing. It should be appreciated that templates that are added to the Template Elements database are independent of the product that required their creation. In other words and by way of example, once one template is created for a European Unity Day Poster, such a template may be globally referenced by the metasystem for use with another poster product for a different market such as an NFL Team poster or a World Cup Soccer Poster.

Once a product template or multiple templates are developed for a particular product line, at the next stage of the invention the Text Translations Module (12) is used to prepare translations for various text elements contained in the Lists dataset that are used by the customer service system either in the displays that are presented to the consumer as part of the custom interface design, or in the creation of the product that is to be marketed to the consumer. As with the templates prepared through the Template Maintenance Module, the need to prepare the translations may have been prompted by the introduction of a new product or product line, but they stand independent of such products and, once specified, may be used globally by the metasystem to accommodate other products or in the development of the display portion of other customer service systems. In the preferred embodiment, for each language specified in the Languages Table, a table of Translation Elements is developed through the Text Translations Module. The storage of text translations on an element basis can be seen in reference to the CopyLangx Table whose data structure is disclosed in Table IV. For example, if the metasystem supports english and spanish translations,

then the system might store an english language element for the word "hello" and a corresponding spanish element for the word "hola". These elements are referenced such that when the language in which the customer service system is operating changes, the corresponding language element is automatically retrieved by the system, thus giving the impression of on-the-fly translation from one language to another.

After the text translations have been entered into the Translations dataset, the Product Planner Module (13) may be invoked. Through this module, all the properties of a product as well as codes associated with the product are assigned in the Product Database (see Figures 3-4). In the first part of the use of the module, the appropriate marketing personnel may specify a number of aspects of the product such as its product number, its description, the occasions for which the product is appropriate, the countries in which the product will appear and the languages the product copy is written in, any licenses under which the product is produced, other products that may be associated with the product and the template associated with the product. Also, the marketing personnel might specify the printer to be used for the production of the product as well as the date the product was first entered into the system. In the second part of this stage, the appropriate creative personnel may determine what copy and designs are to be associated with the product, which, as described below, are supplied as the Design Number and Version contained in the Design Library (24) and the Copy Number and Version from the Copy Library (23). Thus, through the Project Planner Module, a product record is created to designate the basic characteristics of a product. This record, once complete, is then passed to the Product Layout Module (16). As can be seen in Table XVIII, the ProductData database structure brings together a number of parameters

and data to describ the products that can be offered by the customer service system.

If, in the prior stage, neither copy nor a design is available to be assigned, then in the next stages the copy and the design to be associated with a particular product will be stored and organized by the respective Copy Library Maintenance and Design Library Maintenance Modules. Through each of the respective Library Maintenance Modules a unique Copy Number and a unique Design Number is obtained to complete the basic record in the prior stage. Copy identification is stored in the Copy Table whose structure can be found at Table III and a similar table for Designs can be found at Table VII (see also, Figure 5).

It should be noted at this juncture that the Copy Library Maintenance Module (23) also permits the editing of the copy to be included on the product. Copy editors may use this module to divide the copy into logical parts, which constitute elements of the copy. This division permits storage of the copy with its associated customization, if any, and with the customer prompt that initiates the customization sequence with the customer through the Consumer Interface Designer/Kiosk Module (17) as is described in greater detail below. Text translations and text personalization elements ("CopyCustomization") are also developed through the Copy Library Maintenance Module. These items are stored in the metasystem database in the structures set forth in Tables IV and V. As can be seen from each table, the text elements (copylangx) and the personalizations are associated using a compound key with the appropriate element in the Copy database.

The Design Library Maintenance Module (24) also permits a limited form of editing in that it allows a design to be comprised of several different art files. Thus, each design record associated with a Design Number and a unique Design Version may include one or more

Design Elements, where each Design Element record includes, among other items, the file name of the Design Element, its file path and a brief description of the Design Element (see Figure 6). The "DesignElements" are stored as shown in Table VIII. An alternative embodiment of the invention would permit the use of the Design Library Maintenance Module in conjunction with commercially available design creation software to create directly the art files, rather than simply to store and manage them.

Once copy and design elements have been associated by the Product Planner Module, in the next stage the Product Layout Module (16) is used to create a final product file for viewing and production. From the Product Database (21), the copy and designs specified for a particular product become available for display in a WYSIWYG ("What you see is what you get") format. Through the use of this module, a layout artist may take each of the copy and design elements and prescribe the size and position of each element, and with respect to the copy elements, may also specify the font, color, justification and rotation of the copy, as well as setting parameters specifying the limits to which a consumer may customize the copy. It is during this stage that the final product design may be assigned to an appropriate printing device for viewing in printed form. The layout artist may optionally have pertinent product statistics printed on the print test version of the product.

At this point, the product specification, design, development, testing and production in electronic form of a new product are complete and are stored within the metasystem. In the next stage of the invention, profiles of the customer service systems (field units) or "kiosks" through which the product is to be marketed are developed through a Profile Maintenance Module (18). This module permits the specification in a Kiosk Profile (25) database of the characteristics of the kiosk or kiosks

through which a customer will be able to obtain a particular product (see Figure 7). Among the items specified in the Kiosk Profile database (25) are the countries, languages, products, markets and categories that the kiosk is set up to support, as well as default language and country. The "Profiles" data structure can be understood by reference to Table XX. Additionally, the Profile Maintenance Module (18) may be used to specify the marketing frames in the consumer interface that will run prior to the initiation of a product selection menu sequence, and the category of products that are to be excluded from display by the kiosk. When the field units are "running" the custom interface script designed by the interface designer, data from the Kiosk profile is used to make certain elements of the interface design active or inactive to the consumer. Thus through the use of the profile, a single generalized interface script can be designed by the interface designer which then adapts to become a custom script at the field-unit level (see Figure 8).

Once the Kiosk Profiles are complete, a Consumer Interface is then designed or modified by the interface designer to accommodate a new product or products. For purposes of illustration only, and not intending to limit the general application of the invention, at this stage the Consumer Interface Design/Kiosk Module (17) is used by the interface designer to design and maintain a touch screen-based kiosk Consumer Interface for the marketing of the products previously specified and developed through the other modules. In general, the module is designed to permit one Consumer Interface design to serve all markets, languages, demographics, times and geographic regions. This module is specifically designed to permit "on-the-fly" translation and reconfiguration of customer touch screen options, keyboard layout, product selection preferences, blocks of copy used in the product

designs, as well as other elements associated with each interface design.

To serve as a dynamic, flexible marketing tool, the module permits variations in what is displayed to the consumer through a series of "frames", each frame including a number of "smart" controls -- visual and nonvisual elements that are individually programmed to appear or disappear or to be active or inactive at a certain time or in a certain sequence and with which may be associated functions to be performed when the control is actuated. Figures 13 - 22, which are described more fully below, illustrate in greater detail, by way of a flow chart, how an interface designer would interact with the Consumer Interface Designer/Kiosk Module to create such a dynamic customer interface.

Types of "smart" controls that may be placed on a touch screen frame include pictures (such as static graphic files, animation or videos), smart buttons, corrals (which define the spaces in which certain smart buttons or lists of text may appear), keyboards and text. Other controls which may be associated with a particular frame or frames or other controls include vending, sound and timing controls and product display controls. Each control is "smart" in that the function it performs or the manner or sequence in which the function is performed may be made dependent upon the kiosk profile associated with a particular kiosk where the interface is employed, and may further be made dependant upon other parameters such as the current time of year in which the kiosk is operating, a customer selection of an alternate language in which the kiosk can operate, or upon accumulated statistics of customer preferences for a particular product or service. In this manner, for example, the same interface may be used in the United States to display products associated with the Fourth of July, and yet alter itself based upon a different Kiosk Profile to enable French consumers to obtain products relating to

Bastille Day or to display Cinco de Mayo products to Mexican consumers.

This ability to "mutate" is a key feature of the Consumer Interfaces capable of being designed by the Consumer Interface Designer/Kiosk Module. As will be described more fully below, the various controls can be programmed through the module to appear or become active when they are needed and to disappear or remain dormant when they are not. Thus, by way of example, a button for the display of Thanksgiving products may be programmed to appear if a kiosk profile specifies a market that includes Canadian and American consumers but to remain invisible in markets in which Thanksgiving is not a holiday that is celebrated. Extending this example further, the profile of a kiosk in Quebec may specify the display of Thanksgiving products in both French and English up through the October celebration of Canadian Thanksgiving, in accordance with which a smart button will know to appear at a specified time before the celebration, but to disappear immediately after the holiday. A similar kiosk in Buffalo, New York might have a profile that specifies a relevant market for both American and Canadian Thanksgiving products for both French and English-speaking recipients. Thus, for such a kiosk, a smart button would appear prior to the Canadian Thanksgiving that permits the consumer to select Canadian Thanksgiving products, and a few weeks later another smart button would appear to permit the selection of American Thanksgiving products. After the Canadian Thanksgiving has passed, the first smart button would disappear, but the second smart button would remain until the American Thanksgiving Day after which it would similarly disappear.

As previously indicated, sound and animation elements may be associated with a particular frame or control in order to attract consumers or to make the product selection process more interesting. Thus, by way

of example, a Fourth of July smart button, when depressed, may initiate the sounds of fireworks accompanied by a series of animated explosions.

After a customer interface has been designed to accommodate a product line or a new market, the customer interface may be tested in either the design mode (in which all controls, including sound and timer controls, are displayed, regardless of the controls' programmed visibility parameters) or the test mode (in which only controls that are meant to be visible are displayed). Operation in the test mode emulates how the interface will appear to a consumer when interacting with the Kiosk version of the Consumer Interface Designer/Kiosk module which resides in the field units.

After a customer interface is tested, it is then installed in the field units along with the Kiosk version of the Consumer Interface Designer/Kiosk module as well as elements of some of the other modules to provide an on-site vending capability. This resultant on-site customer service system is essentially defined by the kiosk profile in combination with the database of products that may be retrieved through the kiosk-specific customer interface, wherein the interface has an ability to dynamically change the programmed elements in response to (1) the kiosk profile data, (2) changes in the marketing conditions that the kiosk operates in, or (3) changes in the products or services to be offered in response to consumer interaction with the interface.

The operation of the Consumer Interface Designer/Kiosk module can be best understood by reference to Figures 13-22, which describe in flow chart form the method by which an interface designer or "author" develops, tests and executes the customer interface design. This same flow chart also teaches how the Kiosk version of the module plays the interface script to a customer interacting with the customer service system.

Figure 13 details the initialization and setup of the system executing the Consumer Interface Designer/Kiosk module. In steps 140-142, a kiosk profile (141) is loaded from memory store 141 to be used when the system is operating in test or consumer modes. Other global data parameters such as indicated in store 143 are also loaded in this initialization phase. The module then determines whether vending is enabled (145), and if so initializes the vending equipment that is included in the customer service system. If the module is operating in design or test mode on a test and development station, no vending equipment will be attached to the workstation, so step 144 is skipped. At step 146 the current date and time are loaded into the module. At this stage in the program flow a key decision is made which bifurcates the operation of the module between interface design and consumer operation. At step 147 the module determines whether the system is in consumer mode. The consumer mode is active when the module is being operated in a field unit or kiosk. If the system is in consumer mode then control is directed to step 151 which sets up the consumer template by loading data from store kiosk.dsk (149) which contains start-up and configuration data such as what frame is to be loaded first for display to the consumer. Following this configuration step, the first frame to be displayed is loaded at step 153, as defined by the data obtained from store 149. Following the loading of the first frame, the module loops between steps 154 and 160 while loading all of the additional frames that were designed as part of the interface script to be played by the customer. When all the designed frames are loaded from the frame archive (160) control is passed to step 150 which sets up the desktop display that is used by the consumer to interact with the customer service system, and also is used by the interface designer to design custom interfaces, as described below. If in step 147 the module determines that the system is

not in consumer mode, the design template data is loaded into the module from store 148, and control is passed to step 150 for desktop setup. In design or test mode the various tools, titles, menus and borders which surround a frame are displayed to the interface designer, in distinction with consumer mode where such frame controls are not visible.

Figure 14 describes the desktop setup that is engaged during both interface design and customer interaction. This figure details the main control loop of the Consumer Interface Designer/Kiosk module. Starting at node 5A, the module loads any frames which may be present in the frame archive 160. Note that in consumer mode this loop has already been carried out in step 154, so control will immediately pass to step 155. In design and test mode step 155 permits the loading of a frame archive or interface script that is being developed by the interface designer.

If the system is in consumer mode, the outcome of steps 155, 156 and 159 will be negative, resulting in the system being in a loop where steps 163, 164 and 165 process whether a customer has touched or clicked a control, whether a timer control has been activated, or whether there is a sound to be played. As described in more detail below, if a control has been touched or clicked, a process is engaged which operates based on the control activated as set forth in figures 20-22. When a timer is activated as determined by step 164, a timer message is broadcast by the module, and if at step 165 the system determines a sound is to be played, control is passed to a process which plays the appropriate sound (167). Within this main loop the consumer essentially plays the interface script. By touching or clicking on the various controls described in figures 20-22, the customer can select, customize and purchase a particular product offered by the customer service system by

navigating through the frames which form the interface script.

If the system is not in consumer mode, it must be in either design or test mode. In design mode an interface designer is actually designing the various frames which will make up the interface script played when the customer service system is in consumer mode.

In test mode the script is executed just as in consumer mode, but additional controls and visible items are displayed to the interface designer. If the system is in test mode, then at step 156 control is passed to step 159, where the interface script configuration can be saved to store 149. When in test mode, the script itself can be played, and it will appear as it would to a consumer, with all the visibility and activation parameters associated with each control element being enabled.

If the system is in design mode, the module must process additional menu and toolbar selections made by the interface designer. These selections provide the capability and funtionality that the interface designer needs to design, edit, program and configure the various frames that will form the interface script. Steps 157 and 158 determine whether a designer has made a menu or toolbar selection, after which control is passed to step 170 (Figures 16-19), which provides processing for the selected menu or toolbar command. After processing the selected command, control is returned to the main processing loop at node 5A.

Figure 15 provides detail on how the Consumer Interface Desginer/Kiosk module loads a frame archive into memory for editing by the interface desginer or playing by the customer. The archive includes each frame that is part of the interface script, along with the child controls that are placed within each frame, each frame being stored as a persistent object, as provided by the MFC classes. For each frame loaded as part of the

archive the process steps set forth at 171-186 are executed. In step 171 the first control element for a given frame is loaded. The module then loops between steps 172-174 while loading and intializing all remaining controls and sub-controls (or child controls) associated with each parent control. An example of a control with sub-controls is a keyboard control, which is comprised of a number of key controls which form the keyboard. After all the controls are loaded for a given frame, the module loops between steps 176 and 183, while determining, based on steps 176-179, whether a control is to be displayed or not by testing the control's programmed visibility parameters. Once all controls are loaded into memory, the module finally checks whether the frame contains a timer control at step 184, and thereafter returns to the archive load module to load the remaining frames of the interface script.

Figures 16-19 set forth the looping structure that the module traverses through in step 170 of Figure 14, which is responsive to events where the interface designer makes a menu or toolbar command selection. Each decision box merely tests the command selection and directs program control to the appropriate process based on the selection. For example, if the interface designer decides to open a frame on which to place design and control elements, the decision flow stops at step 1710, and program control is transferred to a process at 1711 which prompts the designer for the name of the frame to open, and loads the frame according to the process detailed in Figure 15. Program control is then returned to the main loop in order to process additional commands. Without describing each command in detail, it is apparent from Figures 16-19 what types of commands and tools are available to the interface designer as he authors the interface script.

Figures 20-22 describe the various process actions that a customer or an interface designer can invoke while

the interface script is running. These actions are invoked by selecting certain controls which have been placed by the interface desginer within the various frames that form the script. Starting with step 1810, all the way through 2880, this part of the module merely determines what type of action is to be taken and directs processing to the appropriate method for carrying out that process. For example, in 1810, the module determines whether a control has a sound associated with it, and if clicked engages the appropriate sound file. Most of the processes that can be attached to a control are self-explained by referring to the flow chart itself, for example, moving to a particular frame (1850-1870), executing a program (1880), displaying a particular product (2810-2820), selecting a customization field (2830-2840), selecting a product category (2850), or printing a product (2860). Two processes that require some further explanation are the Change Country process (1830) and the Change Language process (1840). If a button control is clicked or selected which changes the country in which the customer interface is operating, the new country profile data is stored, and a country change operation is broadcast to all controls. Those "smart" controls which are responsive to changes in the country of operation will then automatically take action based on the broadcast message. Similarly if a button control is selected which changes the language in which the customer interface is operating, the new language profile data is stored, a new translation set is loaded from the translations database store (2040), and a message is broadcast to all controls indicating a language change has occurred. Those "smart" controls which are responsive to changes in the language of operation will then automatically perform a text translation of associated text elements based on the message. This completes the description of the Customer Interface Designer/Kiosk module flowchart of Figures 13-22.

During the operation of the customer interface, the Consumer Interface Designer/Kiosk module collects marketing and point-of-sale ("POS") POS data (26) for use in ensuring that credit or charge card purchases are collected and for analysis of marketing trends developed through interactions with the Consumer Interface. Among the market data collected and stored in the POS database are identifiers of the products sold and an indication of the menu paths through which the consumers found the products purchased. The structure of the market data as stored, the "AccumlatedSales," can be found on Table I.

By way of completeness, the hardware used to run the metasystem is now described. In the preferred embodiment the software runs on several platforms including (1) a test and development system, (2) an Art Work station, and (3) a field unit or kiosk. The test and development system, which is used for administrative setup, product specification input, and customer interface design, includes a 486/33 mhz CPU, 32 MB of RAM, a sound card (such as an 8-bit Aztech V.2), a video card (such as an ATI Ultra Pro Mach 32), a CD-ROM drive (with double or triple speed capability) and standard data entry items (such as a mouse, a keyboard and a touch screen). For the Art Work Stations, which are used to layout and design the new products, the equipment in the preferred embodiment is the same as the test and development system, except in order to process efficiently the many graphic application demands on the system, the CPU is preferably a 486/66 mhz or faster CPU. As to the field (Kiosk) units, each unit preferably includes each of the items associated with the test and development systems; however, only 16 MB is required for the preferred embodiment of the field unit parts of the invention. In addition, each test and development system, Art Work Station and field unit will have associated with it printers or plotters (such as the HP DeskJet 1200C, the HP 7500B Plotter or the HP DesignJet 650C).

The heart of the invention is the system by which customer interfaces can be developed or "authored" by the interface designer, as described above particularly through the description of the Consumer Interface Designer/Kiosk module flow charts (Figures 13-22). This component of the overall invention can also be described in reference to the component objects or "controls" (as such objects are preferable termed by the inventors of the metasystem) that are made available to an interface designer, or author, from which to create a dynamic interface that is responsive to changes in the state of the environment in which the interface is "played." Such changes in state, in the preferred embodiment, may be induced by the passage of time, or by the point in time in which the interface is being played, or by a change in the marketing conditions in which the interface is operating, or by consumer interactions with the created interface. It should be easily understood by one of ordinary skill in the art that the controls may also be structured to be sensitive to (and thus responsive to) system errors or unavailability of data (or product), or from the detection by hardware components of the presence of a customer (such as through a voice recognition interface or a motion sensor). In the vending environment, the ability of an interface to mutate in recognition of the unavailability of a particular product (such as through a depleted on-site inventory) is an important feature.

In the preferred embodiment, each control is an instantiation of a specialized window object created in a Microsoft® Windows™ environment through the use of an object-oriented programming language such as VisualC++ or VisualBasic. Each control serves as a building block of the interface that the author/interface designer using the system is trying to create. The type of controls that are available to the author, each with their own specific uses, abilities and characteristics include:

frames, buttons, text areas, graphics areas, timers, vending enablers, sound players, keyboards, keyboard keys, product display areas, list corrals, customization edit corrals and customization text input areas. Associated with each such control is a corresponding object class. Those classes developed as part of this invention are indicated on Figures 9-12 as "Russ Tech Classes" and those utilized from the Microsoft Foundation Classes are similarly indicated as MFC Library classes.

In its simplest embodiment, the Authoring System provides a set of tools for associating one or more controls with a particular type of parent control, called a frame, for each screen that an author using the system wishes to include in a dynamic interface script. To understand the concept of such an interface script, consider by way of analogy an unusual orchestra that plays pieces of a symphony in an order and a manner that is responsive to audience reaction. The orchestra might, for example, play the overture last on the program as a result of the time of the year or the audience reaction to the prior segment performed and play the 2nd movement as the third movement for similar reasons. Alternatively, the orchestra might play the first movement without a horn section, again as a result of audience response (or lack thereof), or because of the date or time at which the movement is to be played, or because the horn section is unavailable or because the concert hall environment does not permit the playing of horns. Similarly, an interface script associated with an interface created using the metasystem establishes pathways and the manner by which the interface will play at any given point in time and in light of any given history of consumer interaction with the interface.

Much like the orchestra and its written music for the various orchestra sections and movements, the interface "performance" cannot be described *a priori* through a true script, but can only be described in terms

of the controls that are available to enable an interface performance that is dependent upon the environment in which it is played and upon audience (i.e. consumer) response. At the level of describing the system that comprises the invention, it is impossible to describe how an interface will function specifically because so much of the interface functionality is dependent upon the design of the interface author using the system and upon the customer interaction with the interface once it is designed and implemented. In one sense, once an interface is made available to a consumer, the consumer plays an important role in directing or programming the interface through the consumer's response to the information presented through the interface.

In light of the foregoing, it can be appreciated by one of ordinary skill in the art that the system can only be described through an understanding of the role certain controls play in the metasystem. Critical to this understanding is first understanding the role of a frame in the system comprising the invention.

Unlike the other control types, the frame is relatively limited in its abilities, but it occupies the top place in the hierarchy of controls. Essentially, when an author instantiates a frame, he creates a canvas upon which the author can place other controls that provide the ability to change the state of the environment in which the interface is designed to run and the pathing of the interface as it plays. These other controls are, by their placement on the frame, considered children of the frame, and the frame is considered the parent object with respect to the other controls. Each frame is defined by a frame class (CVKFrame) (46730) that is similar to, but not derived from the Microsoft Foundation Class (MFC) CFrame. Each class discussed herein can be found on the Authoring System (metasystem) Class hierarchy set forth on Figures 9-12 . For a description of the MFC hierarchy and the class library,

the reader is referred to "Microsoft Foundation Class Library" by Steven Holzer, published by Brady Publishing, copyright 1993. Using the following logical description of the class hierarchy, the detailed flow chart and description of the Consumer Interface Designer/Kiosk module set forth above, and the other diagrams and description of the metasystem contained herein, a programmer of ordinary skill in the art of object oriented design and implementation would be able to code the metasystem of the present invention for use in designing custom interfaces for customer service systems.

When an instantiation of a frame is saved, it is saved as an archive along with its associated children controls. At the beginning of an interface script, a previously-designated starting frame is retrieved from the archive and each of its associated controls becomes "live," i.e., each control can, depending upon its nature, (i) affect what the interface displays (or in the case of sounds, plays) or which frame the interface will play next, or (ii) accept and process input through a touch-screen, a mouse, a keyboard or other input device, including a vending device. The frame itself, however, is incapable of performing any of these functions without having placed upon it (and thus associated with it) one of the other control types. Subsequently other frames can be displayed with their associated control elements. Just exactly which frames are selected for display depends on the interaction with the customer.

Chief among the other controls, and arguably the most versatile, is the button control. A button is an instantiation of the button control class defined in the preferred embodiment by an object class definition, CVKButton (4691). CVKButton, has similarities to, but enables greater functional capabilities than, the MFC button class, CButton. As seen in Figure 9, CVKButton is derived from another class developed by the inventors, which in turn is derived from MFC CWnd. As would be

appreciated by one of ordinary skill in the art of object oriented programming, particularly programming using MFC's, each class that is represented on Figures 9-12 inherits from the class above it, all of the characteristics of that class as defined in the Microsoft Foundation Classes. Thus, the class that governs the control CVKButton is also a CWnd class (461), and inherits all that defines that class.

The key distinction between CVKButton and its parent class is the added "intelligence" that the class provides buttons. Indeed, it is the smart button features of the CVKButton class that enable the system to mutate in response to the passage of time or the state of the system as a result of consumer inputs to the interface or through a system profile associated with the environment in which the interface is designed to operate. As discussed previously, the profile predefines certain initial environmental operating conditions, such as the default language of the interface (U.S.-English, Spanish, Latin American Spanish, German, Italian, UK-English, etc.), the default country of operation, the languages, countries, markets and products the interface supports, the initial date of the operation (month, day and year) of the interface, the serial number of the location where the interface will be installed, the address of the location, an indication of whether the system accepts cash vending and whether the system accepts credit card vending, a pointer to the directory in which environment-specific marketing frames are stored, and a list of those categories of products that, although otherwise available through the interface, are to be excluded from presentation by the interface (see Figure 7). In the preferred embodiment, a button can alter a number of these parameters as will be seen below. It will be appreciated by one of ordinary skill in the art that a button class could be defined that would enable each of the profile aspects to be altered. It should also be

appreciated that an interface could be developed without having a profile at all.

In the preferred embodiment, an instantiation of a button occurs when a button tool is clicked upon in the Consumer Interface Designer module and a button window is drawn upon a frame. Through a dialog box, the button can be "programmed" to perform several tasks and to have certain characteristics. The tasks and characteristics of a button may include:

1. The ability to process that it has been touched (on a touch screen) or clicked (using a mouse), and in response to being clicked or touched, to take certain actions, including:

   a. Transferring control to specific frame;

   b. Executing a program by calling the program;

   c. Changing the operational language of the interface;

   d. Changing the country of operation; or

   e. Playing a sound file or a video clip by transferring control to an object comprising a particular file associated with a particular sound driver (such as a .wav or MIDI file) or video driver (such as QuickTime for Windows, and MPEG player etc.).

2. Determining whether it is to be active (visible) under the operating conditions existing at the time the frame with which it is associated is invoked by examining its visibility parameters and "and-ing" the parameters with the system states determined from the profile and as a result of the prior interface action. If it determines that it is to be visible, it further looks at the length of time that has been set previously through a dialog box and when invoked displays only for that period of time.

3. Displaying itself in accordance with its specific properties, including what text, pictures and colors are to be displayed when the button is in an upstate and when the button is in a down state and

whether it is a push button, a check box or a radio button.

With respect to changing the language, the button when clicked or touched sends out a message to the system that when calling the database, the pointer to relevant language-sensitive data items has been moved from one column of each data table having such data to a second column reflecting the new language. Effectively, the button serves as a means for switching a pointer from one database table column that is language-dependent to another. By way of example, if a button is set to switch the interface language from English to Spanish, then the new database table columns pointers for Spanish will be broadcast to the system and the system, when retrieving text for the interface will correspondingly look in the proper column for the language text. Programmatically, this is accomplished through having each button sense whether it is to change the language when touched or clicked and then broadcasting a message to the system to indicate the pointers that have changed, if any.

Certain other aspects of the Consumer Interface Designer Module which forms the heart of the metasytem and the controls which can be instantiated as part of this module can be understood by further reference to the class hierarchy of Figures 9-12. The object oriented classes specifically designed for use with the invention will be defined and discussed below. The MFC classes are publicly available and will therefore not be defined herein. Using these class definitions, the class hierarchy diagrams, and the rest of the disclosure, figures, flow charts, and descriptions contained herein, a programmer of ordinary skill in the art of object oriented programming could code the metasystem.

The first class developed as part of the Consumer Interface Designer Module is CSQLException (311), set forth on Figure 9. This class is a specialized version of the MFC class CExeption for handling exceptions as a

result of using a Structured Query Language (SQL) database access program. In the preferred embodiment, any ODBC-compliant database management system (such as Microsoft SQL Server, Sybase, DB2, Oracle or Watcom SQL) can be used for database access. Such a database could include data related to global elements, copy, designs, products, or system profiles.

Another class, CMemHugeFile (321) enables the metasystem to save very large files as archive or Binary Large Objects (BLOBS) exceeding a 64K BLOB barrier that is inherent in the MFC CMemFile. Both CMemfile and CMemHugeFile are derived from the MFC CFile (320) class.

To enable the display of plotted text on a monitor so that it appears like the output of the text on a plotter, the class CopyDC(3310) is used to provide a specialized device context based upon the MFC CWindow DC(331) and CDC (330) classes. To permit palette data to be standardized under the Rasterized Interchangeable File Format (RIFF) in the invention, the class CRIFFPalette (3410) has been derived from MFC CPalette (341). Two subclasses, CCorelPalette (3411) and CRGB256Pallette (3412), of the CRIFFPalette (3410) class enable the loading of Corel Palette and RGB palettes, respectively into a RIFF Palette.

In the course of developing the preferred embodiment, it became necessary to develop several general classes that are derived broadly from the CObject (300) class and which affect modules other than the Consumer Interface Designer module . These classes, 301-309, provide primarily miscellaneous housekeeping tools to enable the overall system to perform more efficiently. Thus, CPath (301) permits changes in the current working directory and its subclass, CCACPath (302) establishes a common directory path for use by the metasystem. CDesktop (303) saves and restores the state of the Authoring System at any particular point in its operation. This class is particularly useful in starting

the field unit versi n of the metasystem. CRetailProductClient (304) organizes various sales data for printing a sales report relating to kiosk operation. Classes CDesignElement(305) and CCopyElement (306) enable print and display of design elements, text fonts and rotations of text in accordance with layout instructions for a particular product. CWinExec (307) enables the monitoring and execution of programs outside the metasystem, such as maintenance and sales report generating programs. CAnimationFrame (308) enables the metasystem to perform animation as a part of a control included in an interface design. This class will enable an author to specify the pictures used, the length of playing time and the visibility of the control associated with the animation. Associated with this class is the CAnimation (309) class, which essentially serves as a container and manager of CAnimationFrames.

A number of collections or containers also needed to be developed to implement the preferred embodiment. This included CDBByteArray (351) and CDBDWordArray (351), which contain byte arrays and double word arrays if they are in databases. CDatabaseContainer (370) serves as the database manager. Classes 371-375 on Figure 9 provide containers for array of product criteria, Windows messages, printers and frames. A further collection is CCensorData (381) which performs the functions of managing words that are excluded from text customizations and precluding a customer from using censored words in text customizations. In the preferred embodiment, a given product available for display or purchase may have certain text fields associated with the product that are "customizable". This "customization" is determined early in the product planning and production cycle by the product planner or other creative personnel. The concept behind a text customization is that a product is presented to a customer, with certain text fields that can be "filled in" using customer specific data. The

customer registers these specific data using an input device to create a "custom" product for purchase.

Lastly on Figure 9 are a number of classes relating to document architecture. CVKapp (3911) is the class that implements the Consumer Interface Designer/Kiosk Module in both its design mode and its consumer/test modes. CSQLDocument (3921) is a null class for associating together documents, views and frames. The concepts of "documents", "views" and "frames" are well known to one or ordinary skill in the art of object oriented programming, and the reader is referred again to "Microsoft Foundation Class Library" by Steven Holzer, published by Brady Publishing, copyright 1993 for further information. CVKFrameDoc (3922) is a persistent object class that contains all the "smart" controls associated with a frame as children, stores each frame as a separate file and enables the controls to be retrieved with their corresponding frames. CProduct (3923) is a BLOB archive class that holds all data associated with a particular product. CProductDocTemplate (3931) overrides certain features of Windows (such as borders and title bars) to enable frames to appear as a collection of objects on the screen and not as a window. This enables the interface to have the look and feel of prior art DOS-based touch-screen interfaces without losing certain critical features and functions that the Windows environment provides. Essentially the visible container or boundary that surrounds a window and provides descriptive and sizing controls is removed using this class method.

Referring now to Figure 10, to bring about efficiencies in the operation of the SQL database management system, the CSQLDataBase class (411), a CDataBase class (410) that has been particularized to the invention, was created by removing some features of its parent class and overriding other features. Two subclasses, CSQLServer (4110) and CWatcomServer (4111),

are further particularized to use more efficiently Microsoft SQLServer and Watcom SQL, respectively.

The SQLSet (420) class and the classes derived from it (421, 422, 4220-4222, 4224-2229, 4231-4235, 42221-42229, 42231 and 42232) together enable the functioning of the SQL database subsystems in the preferred embodiment. The functions these classes enable include the supplying of unique record numbers, the time-stamping of creators and modifiers of data, and the defining of column names in database tables.

The multimedia classes (430-434, 440) enable the use of certain multimedia functions in a more efficient manner. The CAAPlayer class enables the system to handle AA animation files in FLC/FLI formats. The CSound class and its subclasses, CCDAudio, CCDTrack, CWAVAudio and CSEQAudio enable the handling and playing of sound information from a variety of sound sources.

The control classes are controls that either MFC did not provide at the time of the creation of the preferred embodiment or were not as efficient as those provided by the inventors. CPopUpMessage (4501) displays text at the start of a routine that goes away after a period of time. CSpiderLeg and CSpider (4502 and 4503) are sizing and aligning tools for placing controls on frames, and CTimer (4504) is a simple timer function not directly provided by the MFC Library.

Turning now to Figure 11, the Dialog Boxes Classes derived from the MFC CDialog (4610) are those classes that generate dialog boxes for allowing interactions with the interface designer, other users of the metasystem involved in the product development process, or in interactions directly with consumers. Of particular interest is CCACFile Dialog (4611), which restricts the selection of files to the pathway directed by CCACPath (302). The remaining subclasses (4612-4634) provide specific dialog box control for inputing data relating to text, passwords, visibility of controls, profile

information, frame information, vending data, product data, edit information, keyboard layout, individual key information, as well as dialog box control for setting other attributes of various controls to be placed within a frame, such as buttons and timers.

The Control Classes (4640, 4650, 4660 and the subclasses under each) implement specialized list box functions, combo box functions and edit functions. Of particular interest are the CDragDrop class (4641), which extends the MFC drag and drop function from files (as supported in Windows 3.1) to generic datatypes and the CAnimationListBox (46419), which allows the selection of animation as part of the interface design. The list box functions set forth in the subclasses (4641 - 46419) under the MFC class CListBox (4640) provide a list box function for such data items as available products, occasions, product styles, paper types, printer types, countries, licensed products, and languages. Also included are list boxes for available copy, designs, products, translations and design elements. Similar specialized functionality is set forth in the combo box subclasses (4651 - 465130) for implementing a combo box function.

A frame is the basic window that elements, including text, controls, etc., are placed on by the interface designer. MFC class CFrameWnd (4670) is the parent class from which frame subclasses specific to the present invention derive their functionality. Under CFrameWnd are the two main classes for defining a parent frame, CMDIFrameWnd (4671) which is a frame at the highest level in a frame heirarchy, and CMDIChildWnd (4673) which defines the attributes of a child frame or sub-frame which exists within the parent frame. The CMDIProductFrame (46710) is a parent frame used to display a particular product to the interface designer to be incorporated into the interface script, whereas the CMDIRetailProductFrame (46711) which is a subclass of the

CMDIProductFrame is used to display the same product frame when the Consumer Interface Designer Module is being played by a customer. Certain aspects of the frame which are visible to the interface designer through CMDIProductFrame are not visible to the customer operating the field-unit. The CVKMainFrame (46721) and CVKConsumerMainFrame (46722) are the primary frame classes upon which the various controls and elements are placed by the interface designer, and interacted with by the consumer. The difference between the two is that certain aspects of the CVKMainFrame, such as title bar, border outline and sizing control, which are visible to the interface designer, are not visible to the customer who interacts with frames instantiated using the CVKConsumerMainFrame class. Subclasses 46730 - 46741 are used to instantiate child objects which are essentially sub-frames that are placed by the interface designer within the main frame, and inherit functionality from the CMDIChildWnd (4673) MFC class. The CVKFrame (46730) is the general child frame class upon which the various controls and elements are placed by the interface designer. Under CVKFrame are three sub-classes, CVKTestFrame (46731), CVKDesignFrame (46732) and CVKConsumerFrame (46733). The test frame class defines a frame being displayed in the test mode, whereas the design frame class defines the attributes of a frame being used in design mode, and finally the consumer frame class defines the attributes and functionality of a frame being played to a customer in consumer or retail mode. The differences between the attributes associated with a design frame and a consumer or test frame are similar to those associated with the CVKMainFrame and CVKConsumerMainFrame. In design mode the various elements, buttons, text, etc., are placed into the frame by the interface designer, during which time all visibility or activation parameters associated with a given control are inactive. In the test mode, the

visibility controls which determine whether a given element is active or inactive are operative, such that certain elements may not be visible depending on their control settings. In this mode, like in design mode, the title bar, window border, scroll bars and sizing tool will be visible. The consumer frame class defines all visibility attributes to be active, like the test mode, but the window title bar, border, etc. are not visible. There is also a difference in display resolution between CVKConsumerFrame, which uses an 800x600 pixel display and CVKTestFrame and CVKDesignFrame which display in 1024x768 resolution. CProductFrame (46740) and CProductFrameRetail (46741) are child frame classes which enable multiple product frames to be displayed within a CVKMainFrame instantiated object.

A view is the area within a frame that data, product, text, elements and controls are placed within. In MFC parlance this combination of elements placed within a view is known as a document. The frame includes at least one view, but also provides the sizing, title, scrolling and other window controls which exist outside of the view. The specialized view classes (46810-46821) which form a part of the preferred embodiment provide the basic functionality to integrate the view into a frame as described in the preceding paragraph. For example, the CProductView (46820) is used with the CProductFrame (46740), and the CVKDesignView is used with the CVKDesignFrame, and so on.

The core classes of the preferred embodiment are defined by the control sub-classes set forth under CVKControl (4690). These classes define each of the "smart" controls discussed above that can be place within a frame by the interface designer. Each of the controls is "smart" because the attributes which define the various control classes permit information flow to the controls upon which decisions are made related to visibility and activation. Information such as time of

year, marketing conditions, languages selected, etc., can flow to the "smart" control classes so that the controls can become active or inactive depending on this information.

The CVKButton class (4691) has all of the general attributes of a regular windows button, with the added functionality described above which makes the controls "smart". While most controls use the standard CVKButton, several specialized buttons have been devised as part of the preferred embodiment, including CVKListButton (46911) which is programmed with attributes that enable lists of information to be properly displayed in a list corral, CVKFixitButton (46912), which is a specialized button which contains methods for placing a customizable message in a product, CVKVend (46913), which if activated, controls the vending processing by, for example, prompting the customer for monetary or credit card input prior to dispensing the customized product, and CVKKey (46914), which is a button class programmed with information on the scan code for a given key, such that when the button is displayed, no matter what language the customer service system is operating in, the button will display the correct letter by comparing its programmed scan code with a stored look up table which translates scan codes into language characters.

The CVKTimer class (4692) is a temporal control button which can be set to do something in a specified period of time. It can be set to play a sound, or display a picture, direct program control to a different frame, etc., at the expiration of some time period. This is a basic control that the interface designer can use to control the timing of the interface script which is being played by the consumer.

CVKPic (4694) is the picture control that operates with graphics or animation files which provides the interface designer with the functionality to place,

process and display graphical objects and full-motion animation as part of the interface script.

Another control heirarchy is based on the CVKCorral class (4694). The CVKCorral class is not used to instantiate objects, but rather is an abstract class from which the subclasses CVKCorralList (46940) and CVKCustomList (46941) inherit functionality. CVKCorralList is used to take a specified list of buttons or controls and place them within the space specified by the interface designer, automatically setting the spacing and size of the controls in the associated list. The Corral is a control which is comprised of a series of sub-controls.

The CVKCard (4695) control class is used to instantiate an object which sets forth the display of the product, this classes attributes include control of the sizing of the product, as well as any text to be displayed along with the product. CVKSound (4696) is a control class with attributes that allow the playing of sound files, such as a "wave" or "midi" files. CVKText (4697) is a control class which allows intantiation of text objects which are just variable text fields which can contain any type of text. The CVKText control class, like all other "smart" control classes, includes attributes which allow instantiated objects of the class to become dynamically active or inactive depending on the setting of the visibility controls by the interface designer. The CVKKeybd class defines the keyboard unit into which individual keys are placed, and operates in similar fashion to the CVKCorralList (46940) function in that it is a control that is comprised of a set of sub-controls. In the case of the CVKKeybd class, it is comprised of CVKKey (46914) control objects. The final control class is the CVKEdit (4699) class, which has attributes which control the customization of text by a customer, the CVKEdit class being able to determine whether too much or too little text has been input by

using stored data entered previously as part of the Product Layout Module.

Turning lastly to Figure 12, there are set forth six additional groupings of classes which have been custom coded in support of the present invention, Product Support (520-590), RT Classes (610-770), Visual Kiosk Profiles (780-830), Graphic Support (840-870), Line List Support (880-890), and Vending Support (910-970). Also included on Figure 12 is the CDBTimeStamp class (511), which is derived from the MFC CTime class. This class contains the functionality which allows a formatted time and date stamp to be entered along with product or global data input into the various databases and also includes attributes which allow the identity of the person making the entries or changes to be stored in the database.

The Product Support grouping 520-590 contains classes which support the presentation and customization of the products to be offered by the customer service system authored using the metasystem. Starting with CTextTran (520), this class provides the functionality which enables proper text translation among a variety of languages by obtaining the proper record stored in the Copy Library files. CCensorShip (530) is the class that a listbox would call to check whether certain data items should be excluded from the list based on a programmed censorship level. The preferred embodiment of the present invention envisions a number of censorship levels (or Tier Levels), similar to a motion picture rating scheme, where a product could be classified as G-rated all the way through X-rated where appropriate. CCustomizer (540) is the class that processes a customizable text element by knowing when and where to prompt the customer for custom text input into the customizable element. The CKeyboard class (550) is the key and keyboard controller which knows how to keep a physical keyboard, if one exists, and a displayed keyboard on a touch screen in sync with one another. It

essentially mimics a physical keyboard input when the user of the metasystem is actually providing inputs through a touch screen display of a keyboard. CTokenParser (560) is a string handler class which handles tokens which could, for example, exist within a customizable text element, the tokens being inserted by a product designer to indicate where the custom text is to be inserted by the consumer. The CTokenParser class operates to parse out or identify the location of such tokens for further processing. CProductTemplate (570) is the class which retrieves the current template for a given product, and CProductComponent (580) is the class which ties the design and copy of a particular product together to form an integrated unit. The final Product Support group class is the CMutator class (590), which is used during the display of copy and adds the proper control characters to the copy such that the copy is displayed in the proper size, color, and format.

The classes identified as part of the RT Classes grouping are specialized classes which perform functions required by the preferred embodiment, but which are not part of any general category or definition. These are miscellaneous classes which were designed to meet various functional requirements of the metasystem. The CCursorExcursion (610) class provides a memory of cursor type and position such that after an excursion or movement is made which alters the cursor type or location, the class can automatically return the cursor to its previous state. CExceptionHandler (620) handles exceptions which may occur within any process. CFieldExchange (630) handles the data exchange between a specific database format, such as a numeric or date format, and a C++ or C format such as a string or integer. The CFileDlg class (640) requires that a design or copy element that is being placed into a frame is only retrievable from a specific directory location of the system's permanent storage. The CInteruptHandler (650)

is a process class which traps any General Protection Fault errors which may occur when the Customer Interface Designer/Kiosk module is operating in the retail mode, and may take some action, such as rebooting the system. CKioskLog (660) is a class which stores certain event notifications that a service person who is maintaining the customer service system can use to audit what events have occurred to a particular unit, such as if the date/time of the unit was changed, or how many times the unit was "booted" every day. CMath (670) provides for additional floating point processing which is used for certain graphics operations performed by the preferred embodiment. CMFCTrace (680) is a class which enhances debugging of the software modules of the present invention, and is only utilized in testing the software modules during and after coding. The CMouseCursor class (690) provides functionality which handles processing of which cursor type is displayed when the mouse is at a given position. The COwnerDrawEntry class (710) provides enhanced functionality to the standard listbox controls which permit the addition of specialized graphics or other elements into the listbox. CFrameManagerEntry (720) is a class which manages the entry of frames and sub-frames by the interface designer into an interface script. CVKDefaults (730) is a class which stores all the initial program defaults for the metasystem. CVKTrace (740) is a class which allows the interface designer to specify a specific frame as a starting point as part of a test or design process for a particular interface script. CSpiderWeb (750) is a class which handles the resizing box on a given window that is capable of being resized. The CPrinterManager (760) is a class which handles what printer is being utilized to print a customized product and handles the configuration setup for a particular printer type. The final class in the generalized RT Classes group is the CPaletteExcursion class (770) which remembers a specific palette type and

location and returns control to that particular palette after an excursion is made by the interface designer.

Another class grouping on Figure 12 which adds additional functionality to the preferred embodiment is the Visual Kiosk Profiles group (780-830). Together these classes provide the activation profile, or visibility parameter settings, for a particular control. Each control is partially defined by its profile which contains each of these classes. This enables the "smart" control to know its visibility, sound attributes, behavior, and activation criteria.

The Graphic Support grouping (840-870) consists of classes which support certain graphical operations. CDib (840) is a class which supports device independent bitmaps. CDrag (850) provides drag and drop support. CGraphics (860) is a generic graphics handler. The CDCExcursion class (870) is another excursion handler similar to those described above which handles excursions from a particular device context.

The CLineList class (880) provides a list which starts with a hierarchy by product and further provides a list of the selected products in a categorized format, each list entry being composed of an object instantiated from the ClineListEntry class (890), which defines the individual product entries.

The final grouping on Figure 12 is the Vending Support group. This group provides general functionality to support the use of vending equipment with the customer service system that is playing the interface developed with the metasystem. CComm (910) provides communications support to the vending device. CVendingEquipment (920) provides the functionality to determine what type of vending equipment, such as coin deposit, bill reader, or credit-card reader, is installed with the customer service system. The CCreditCardValidator (930) is the class which validates a credit card using specific algorithms obtained from the various credit card

companies whose cards are supported by the customer service system. CCreditCardParamter (940) is a class which allows input of data which defines the format of a particular credit card that is to be validated using CCreditCardValidator (930). CCashValues (950) handles the type of currency entered into the vending equipment and converts a currency type signal into its proper cash value depending upon the country that the customer service system is operating in. Finally, the CVend class (980) is the method which handles the vending button calls and handles the communication with the vending equipment. This completes the discussion of the metasystem class hierarchy.

Utilizing the foregoing detailed description of the class hierarchy, in conjunction with the diagrams, flow charts, tables, and description of the preferred embodiment set forth herein, a programmer of ordinary skill in the art of object oriented programming would be able to develop the "Authoring" system that is the present invention.

Although the present invention has been described and illustrated in detail above and through the Figures, it is intended that the spirit and scope of the invention be interpreted as including the foregoing by way of illustration, and that the same be limited only by the appended claims as interpreted in light of the foregoing and all other equivalents.

Authoring System Database (ASDATA) Dictionary

TABLE I
AccumulatedSales - Stores Product Sales Data

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/AccumulatedSales)
FK	SerialNumber	varchar	255	Foreign Key to Profiles
FK	ProductNumber	int	4	Foreign Key to LineList (ProductNumber)
	CreditCardNumber	varchar	255	If vending by credit card, the card number. If vending by cash, "CASH"
	Amount	float	8	If vending, the collected sales amount
	VendType	int	4	0=Cash. For others see Vending Parameter Maintenance
	CDVolume	int	4	CD Volume in the format 'xxmmyy' where xx=Volume Number
FK	Category	int	4	Foreign Key to LineList (RecordNumber)
	Frame	varchar	255	List of the frame names traversed to this product selection
FK	CustomCountry	int	4	Foreign Key to Countries - Country at time of personalization
FK	CustomLanguage	int	4	Foreign Key to Language - Language at time of personalization
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	
	QuantityPrinted	int	4	Number of copies printed

Authoring System Database (ASDATA) Dictionary

TABLE II
BlobData - Stores Archive data (i.e. Line List display)

	DataType	int	4
	Blob	image	16

TABLE III
Copy - Stores Copy identity information 1 per Copy Number

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Copy)
K	CopyNumber	int	4	CopyNumber, never reused (RecordNumbers/CopyNumber(Field))
K	CopyVersion	smallint	2	Copy Version
	Description	varchar	254	Copy Description
	Editor	varchar	254	Copy Editor Name
	Author	varchar	254	Copy Author Name
	Source	varchar	254	Copy Source
FK	License	int	4	Foreign Key to License table
	DateCreated	datetime	8	Time stamp when row was inserted
	CreatedBy	varchar	50	User name (from SYSTEM.ANYWAY, UserName=)
	DateModified	datetime	8	Time stamp when row was inserted, or changed
	ModifiedBy	varchar	50	User name

TABLE IV

CopyLangx - Stores copy text elements by language (x = Languages key). Many per Copy row

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/CopyLangx)
K	CopyNumber	int	4	Combined with Copy Version for compound key to Copy and with ElementNumber to uniquely ID row
K	CopyVersion	smallint	2	
K	ElementNumber	int	4	0 through number of elements within CopyNumber/CopyVersion
	Description	varchar	255	Copy element description
FK's	Element	text	16	The copy element - contains 1) text 2) Foreign key(s) to CopyCustomizations or 3) both
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE V
CopyCustomizations - Stores each unique combination of personalization characteristics ('Precustom' thru 'List')

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/CopyCustomizations)
FK	Precustom	int	4	Foreign Key to Lists
FK	PrecustomList	int	4	Foreign Key to ListxLangy where x=PrecustomList value and y=Languages key
FK	KeyBPrompt	int	4	Foreign Key to Lists
FK	KeyBPromptList	int	4	Foreign Key to ListxLangy where x=KeyBPromptList value and y=Languages key
FK	List	int	4	If not 0 Foreign Key to Lists (List to display i.e. 'Closings' list)
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE VI
Countries - Stores Country Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Countries)
	Text	varchar	255	Country Name
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE VII
Design - Stores Design identity information 1 per Design Number

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Design)
K	DesignNumber	int	4	DesignNumber, never reused (RecordNumbers/DesignNumber(Field))
K	DesignVersion	smallint	2	Design Version
	Description	varchar	255	Design Description
	Editor	varchar	255	Design Editor
	Author	varchar	255	Design Author
	Source	varchar	255	Design Source
FK	License	int	4	Foreign Key to Licenses table
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE VIII

DesignElements - Stores design element path/file name. Many per Design row

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/DesignElements)
K	DesignNumber	int	4	Combined with Design Version for compound key to Design and with ElementNumber to uniquely ID row
K	DesignVersion	smallint	2	
K	ElementNumber	smallint	2	0 through number of elements within DesignNumber/DesignVersion
	Description	varchar	255	Element Description
	Element	text	16	Path and File Name of design element
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE IX
Languages - Stores Language Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Languages)
	Text	varchar	255	Language Description
	KeyboardDLL	varchar	50	Keyboard DLL for this language
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	

TABLE X
Licenses - Stores Licensed Character Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Licenses)
	Text	varchar	255	Licensed character name (i.e. 'Rug Rats')
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	
	Licensor_name	char	40	Licensor Name (i.e. 'Nickelodeon')

TABLE XI
LineList - Stores Line List categories and Product Numbers

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/LineList)
K	Category	int	4	Identifies unique grouping of Product, Occasion, Style, SubCat1-4
K	Sequence	int	4	Product display sequence within category (if =0 it is not a Product)
FK	Product	int	4	Foreign Key to Products table
FK	Occasion	int	4	Foreign Key to Occasions table
FK	Style	int	4	Foreign Key to Styles table
FK	SubCat1	int	4	Foreign Key to SubCategories table
FK	SubCat2	int	4	Foreign Key to SubCategories table
FK	SubCat3	int	4	Foreign Key to SubCategories table
FK	SubCat4	int	4	Foreign Key to SubCategories table
FK	ProductNumber	int	4	Foreign Key to ProductData table
	Type	int	4	For Drag & Drop (T/F)
	Level	int	4	Indentation for Listbox display (0-n)
	Rating	float	8	
	Description	varchar	255	From ProductData for Reporting Level description, 'Text' from lowest level, or Product Description
	DateCreated	datetime	8	
	CreatedBy	char	50	
	DateModified	datetime	8	
	ModifiedBy	char	50	

Authoring System Database (ASDATA) Dictionary

TABLE XII
ListxLangy (x = 'Lists', y = 'Languages') - Stores 'text' by List (Button, Pre-Customization, Keyboard Desc., etc), by Language

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/ListxLangy)
	Text	text	16	Translated Text
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE XIII
Lists - Stores List Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Lists)
	Text	varchar	255	List Description
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	

TABLE XIV
Markets - Stores Market Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Markets)
	Text	varchar	255	Market Description
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	

TABLE XV
Occasions - Stores Occasion Names and notes seasonal or everyday distinction

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Occasions)
	Text	varchar	255	Occasion Description
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	
	seasonal_everyday	char	1	Denotes whether seasonal or every day occasion (E or S)

TABLE XVI

PaperTypes - Stores Paper Type Names and characteristics

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/DesignElements)
	Text	varchar	255	Paper Type Description
	PaperType	int	4	Paper Type (sheet, envelope, etc)
	Width	real	4	Paper Width
	Height	real	4	Paper Height
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE XVII
Printers - Stores Printer Names, Driver Names, Alternate Printer Keys, and Paper Types Used

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/DesignElements)
	Text	varchar	255	Printer Description
FK	PaperTypes	image	16	Array of Paper Types allowed
FK	AlternatePrinters	image	16	Array of Alternate Printers (Keys back to this table)
	DeviceName	varchar	255	Windows Print Device Name
	DriverName	varchar	255	Windows Print Driver Name
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	

TABLE XVIII
ProductData - Store Product Information

PK	RecordNumber	int	4	Unique ProductNumber, never reused (RecordNumbers/ProductData)
	Rating	real	4	Product Rating
FK	CopyNumber	int	4	Foreign Key to
FK	CopyVersion	int	4	Copy table
FK	DesignNumber	int	4	Foreign Key to
FK	DesignVersion	int	4	Design table
	Archive1	image	16	Archive of template information
	Archive2	image	16	Archive of layout data, instructions for putting the product together from the specified copy and design elements. Position, fonts, sizes, rotation, etc.
FK	Countries	image	16	Array of Foreign Key to Countries table - Controls country visibility
FK	Languages	image	16	Array of Foreign Key to Languages table - Controls language visibility
FK	DefaultLanguage	int	4	Foreign Key to Language table - base language of product

TABLE XVIII
ProductData - Store Product Information
(Continued)

FK	Markets	image	16	Array of Foreign Key to Markets table - Controls market visibility
FK	Stock	int	4	Foreign Key to PaperType table
FK	Printer	int	4	Foreign Key to Printer table
FK	Associate	int	4	Foreign Key to ProductData table (i.e. associating an RSVP to an Invitation)
FK	License	int	4	Foreign Key to Licenses table
FK	Template	int	4	Foreign Key to Template table
	Activity	int	4	0 = Active, 1 = Inactive
	ReworkCopy	int	4	1 = Copy to be rework
	ReworkDesign	int	4	1 = Design to be rework
	ReworkAttributes	int	4	1 = Layout to be rework
	Description	varchar	255	Product Description
	AssociatedDescription	varchar	255	Associated Product Description
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	
	TierLevel	int	4	Controls tier visibility

TABLE XIX
Products - Stores Product Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Products)
	Text	varchar	255	Product Description
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	

TABLE XX
Profiles - Stores unique Unit Profile information

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Profiles)
K	SerialNumber	varchar	255	Unit Number ('AA999999' format)
	StoreName	varchar	255	
	Address	varchar	255	
	City	varchar	255	
	State	varchar	255	
	Country	varchar	255	
	Zip	varchar	255	
	Phone	varchar	255	
	DefaultCountry	int	4	Kiosk's Default Country
	DefaultLanguage	int	4	Kiosk's Default Language
	Countries	image	16	Array of Kiosk's available Countries
	Languages	image	16	Array of Kiosk's available Languages
	Products	image	16	Array of Kiosk's available Products
	Market	int	4	Kiosk's Market designation
	Contact	varchar	255	
	MarketingScreen1	varchar	255	Path to special substitute marketing screens (same name as original)
	CustomText	varchar	255	Text for back of product
	DateCreated	datetime	8	
	CreatedBy	varchar	50	

Authoring System Database (ASDATA) Dictionary

DateModified	datetime	8
ModifiedBy	varchar	50

TABLE XX
Profiles - Stores unique Unit Profile information
(Continued)

	CashPrice	float	8	Vending Price for cash units
	CreditPrice	float	8	Vending Price for credit card units
	CashReceipt	int	4	0=Always, 1=Never, 2=Optional
	CreditReceipt	int	4	0=Always, 1=Never, 2=Optional
	TierLevel	int	4	Unit's censorship tier level

TABLE XXI
RecordNumbers - Stores current values for incremented numbers throughout the system

PK	TableName	varchar	50	Table or field name
	CurrentRecord	int	4	Last used number

TABLE XXII
Styles - Stores Line List Style Names

PK	RecordNumber	int	4	Unique number, never reused (RecordNumbers/Styles)
	Text	varchar	255	Style Description
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	

TABLE XXIII
SubCategories - Stores Line List Sub-category 1,2,3,4 Names

PK	RecordNumber	Int	4	Unique number, never reused (RecordNumbers/SubCategories)
	Text	varchar	255	Sub-category Description
	DateCreated	datetime	8	
	CreatedBy	varchar	255	
	DateModified	datetime	8	
	ModifiedBy	varchar	255	

TABLE XXIV
Templates - Stores Product Template data

PK	RecordNumber	Int	4	Unique number, never reused (RecordNumbers/Template)
	PrintLandscape	Int	4	Is this to be printed landscape? 0=Yes, 1=No
	ViewLandscape	Int	4	Is this to be viewed landscape? 0=Yes, 1=No
	Duplex	Int	4	Is this to be duplexed? 0=Yes, 1=No
	Name	varchar	255	Template Name
	Archive	varchar	255	Archive of fold line, mirror line, etc. data
	DateCreated	datetime	8	
	CreatedBy	varchar	50	
	DateModified	datetime	8	
	ModifiedBy	varchar	50	
	UnitsPerSheet	Int	4	Quantity of items per printed page (i.e. invitation = 3)

WHAT IS CLAIMED:

1. A customer service system interface development tool for use by an interface designer in creating an interface for customer interaction, wherein such interface is to be incorporated into a customer service system for presenting products or services to a customer for the customer to make a product or service selection if the customer so desires from the products or services presented as a result of the customer's interaction with the interface, such interface development tool comprising:

a global elements maintainer for enabling the interface designer to specify and maintain global elements associated with the products or services to be presented to the customer and for storing the global elements specified and maintained by the global elements maintainer in a global elements storage area;

a profile maintainer for developing a profile of the customer service system environment in which the interface is to operate and for storing the profile so developed in a profile storage area, wherein such profile maintainer under the direction of the interface designer interacts with the global elements storage area to select a set of profile elements from the global elements and further wherein the profile includes the set of profile elements so selected;

a presentation planner for associating a set of presentation data with the products or services available for presentation to the customer and for storing the set of presentation data in a presentation data storage area, wherein the set of presentation data includes a subset of data that are selected from the global elements; and

an interface developer for developing the interface, the interface comprising a set of one or more presentation frames operating in accordance with the profile associated with the customer service system

environment within which the interface is designed to operate and in conjunction with one or more controls associated with the set of one or more presentation frames, such that when the interface is implemented on the customer service system, activation of one or more of the controls or a particular presentation frame will result in an event, wherein such event may include the activation of another control or presentation frame or display to the customer of presentation data, and wherein at least one control may be activated as a result of action by the customer.

2. A process for creating an interface for customer interaction, wherein such interface is to be incorporated into a customer service system for presenting products or services to a customer for the customer to make a product or service selection if the customer so desires from the products or services presented as a result of the customer's interaction with the interface, such process comprising the steps of:

specifying and maintaining global elements associated with the products or services to be presented to the customer;

storing the global elements in a global elements storage area;

developing a profile of the customer service system environment in which the interface is to operate;

storing the profile so developed in a profile storage area;

associating a set of presentation data with the products or services available for presentation to the customer;

storing the set of presentation data in a presentation data storage area; and

developing an interface comprising a set of one or more presentation frames operating in accordance with the profile such that when the interface is implemented on

the customer service system, activation of one or more of the controls or a particular presentation frame will result in an event, wherein such event may include the activation of another control or presentation frame or display to the customer of presentation data, and wherein at least one control may be activated as a result of action by the customer.

3. The customer service system interface development tool according to claim 1, wherein the customer service system is a point-of-sale or point-of-preview kiosk.

4. The customer service system interface development tool according to claim 1, wherein the customer service system is an in-home interactive system.

5. The customer service system interface development tool according to claim 1, wherein the presentation planner further comprises a production planner for enabling production of customer products in accordance with results from a customer's interaction with the interface and at a location proximate to the customer service system within which the interface is to be incorporated.

6. The customer service system interface development tool according to claim 5, wherein the customer products include greeting cards.

7. The customer service system interface development tool according to claim 6, wherein the customer products include posters.

8. The customer service system interface development tool according to claim 7, wherein the customer products include social occasion announcements.

9. The customer service system interface development tool according to claim 1, wherein the customer service environment includes a touch screen upon which the interface is displayed and through which a customer indicates product or service preferences.

10. The customer service system interface development tool according to claim 9, wherein at least one control is a button located on the touch screen that when depressed by a customer will dynamically change the interface from one language to another.

11. The customer service system interface development tool according to claim 10, wherein at least one control is a button located on the touch screen that when depressed by a customer results in a change in a presentation of the products or services.

12. The customer service system interface development tool according to claim 11, wherein the change in the presentation reflects an interaction of information conveyed by the customer through the touch screen with the profile.

13. The customer service system interface development tool according to claim 11, wherein the change in the presentation of products or services is determined by accumulated statistics collected by the customer service system of consumer preferences for certain types of products or services registered through the button control.

14. The customer service system interface development tool according to claim 1, wherein at least one control to be interacted with by a customer becomes visible or invisible for customer selection based on the time of year in which the customer service system is operating.

15. An int rface produced by the process according to claim 2, wherein the products or services comprises products and services relating to the greeting card industry.

16. An interface produced by the process according to claim 2, wherein the interface is designed to be incorporated as part of a customer service kiosk.

17. A customer service system for permitting a customer to select a product or service for preview or purchase, the customer service system comprising:

a memory for storing product data and system profile data;

a processor for configuring the customer service system in accordance with the profile data stored in the memory;

a display for displaying a customer interface, the customer interface including display elements, said elements comprising frames and controls within each frame, both of which are operatively made active or inactive depending on the system profile data, said customer interface further including a display of products or services available for preview or purchase by the customer;

an input device for allowing a customer to register responses, where appropriate, to controls displayed within the customer interface, said responses including a selection of a particular product for purchase; and

said customer service system including an ability to dynamically change at least a portion of the displayed elements, or products and services, based upon changed marketing conditions or certain customer responses registered through the input device.

18. A customer service system for permitting a customer to select a product or service for preview or purchase, the customer service system comprising:

a memory for storing product data and system profile data;

a processor for configuring the customer service system in accordance with the profile data stored in the memory;

a display for displaying a customer interface, the customer interface including frames and controls within each frame, both of which are operatively made active or inactive depending on the system profile data; said customer interface including a display of products or services available for preview or purchase by the customer;

an input device for allowing a customer to register responses, where appropriate, to controls displayed within the customer interface; and

mutation means for dynamically changing at least a portion of the displayed frames or controls, or products and services, based upon changed marketing conditions or certain customer responses registered through the input device.

19. In a customer service system designed to allow a customer to select a product or service for preview or purchase, said customer service system including a memory for storing product data and system profile data, a processor for configuring the system according to the profile data, a display, and an input device for registering customer responses, the customer service system further comprising:

a customer interface to be presented to the customer via the display, said interface comprising frames, said frames including text elements and control elements, wherein the frames, text elements and control elements become active or inactive in response to the system profile data;

said customer interface including a display of products or services available for preview or purchase by the customer; and

said customer interface further including a means for dynamically changing the display of frames, text elements, control elements, and products and services available for preview or purchase, based upon changed marketing conditions or certain customer responses registered through the input device.

20. The customer service system according to claim 17, futher comprising:

a printer for printing the product that is selected by the customer for purchase.

21. The customer service system according to claim 17, wherein the changed marketing condition which causes a dynamic change in at least a portion of the displayed elements is the time of year in which the customer service system is operating.

22. The customer service system according to claim 17, wherein the changed marketing condition which causes a dynamic change in at least a portion of the displayed elements is the language in which the customer service system is to display the interface.

23. The customer service system according to claim 17, wherein the customer responses registered through the input device which cause a dynamic change in at least a portion of the displayed elements, or products and services, include accumulated statistics recorded by the customer service system on which products and services have been most frequently selected by other consumers.

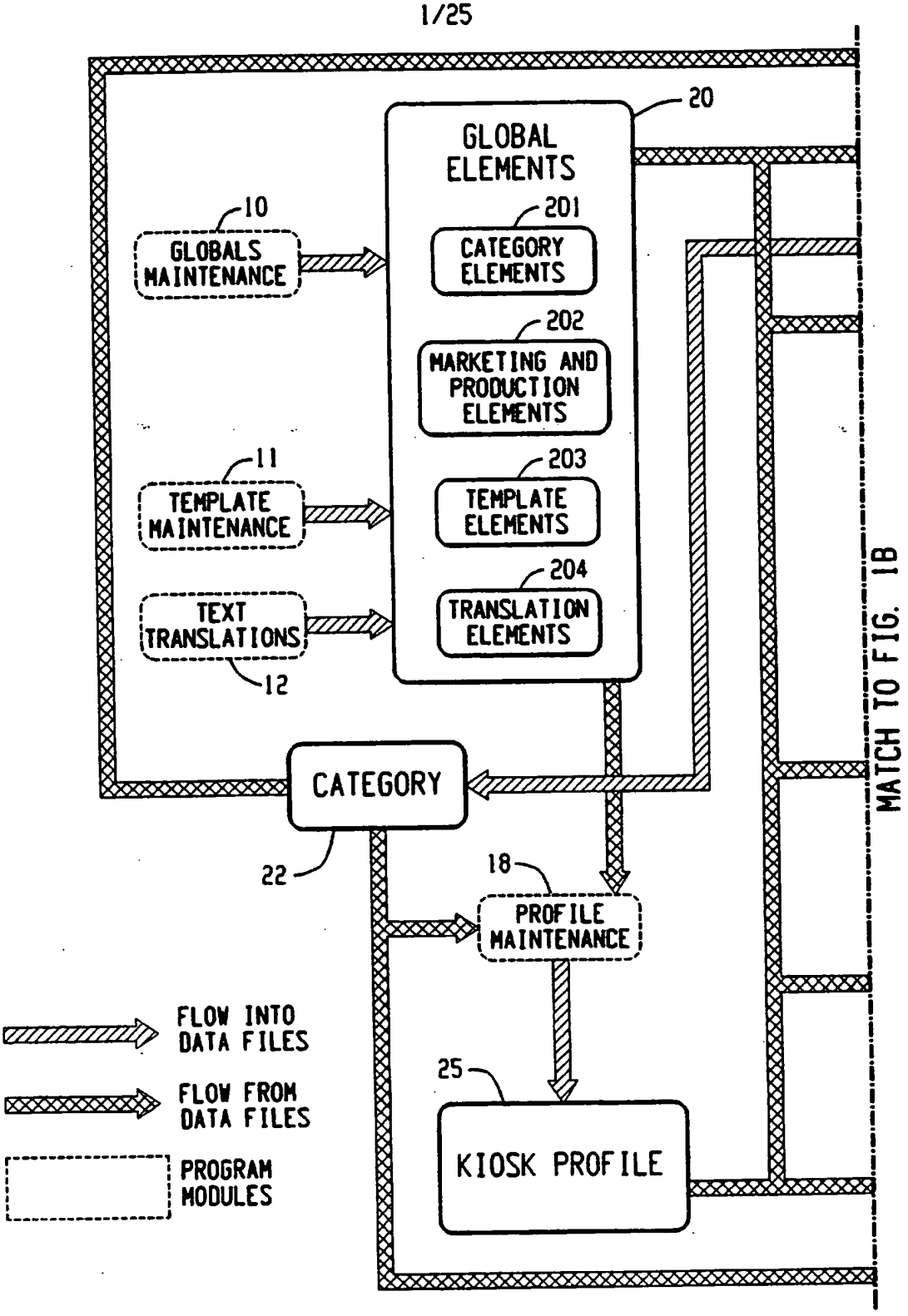

GLOBAL ELEMENTS
20
GLOBALS MAINTENANCE
10
CATEGORY ELEMENTS
201
MARKETING AND PRODUCTION ELEMENTS
202
TEMPLATE MAINTENANCE
11
TEMPLATE ELEMENTS
203
TEXT TRANSLATIONS
12
TRANSLATION ELEMENTS
204
CATEGORY
22
PROFILE MAINTENANCE
18
KIOSK PROFILE
25
MATCH TO FIG. 1B
FLOW INTO DATA FILES
FLOW FROM DATA FILES
PROGRAM MODULES

FIG. 1A

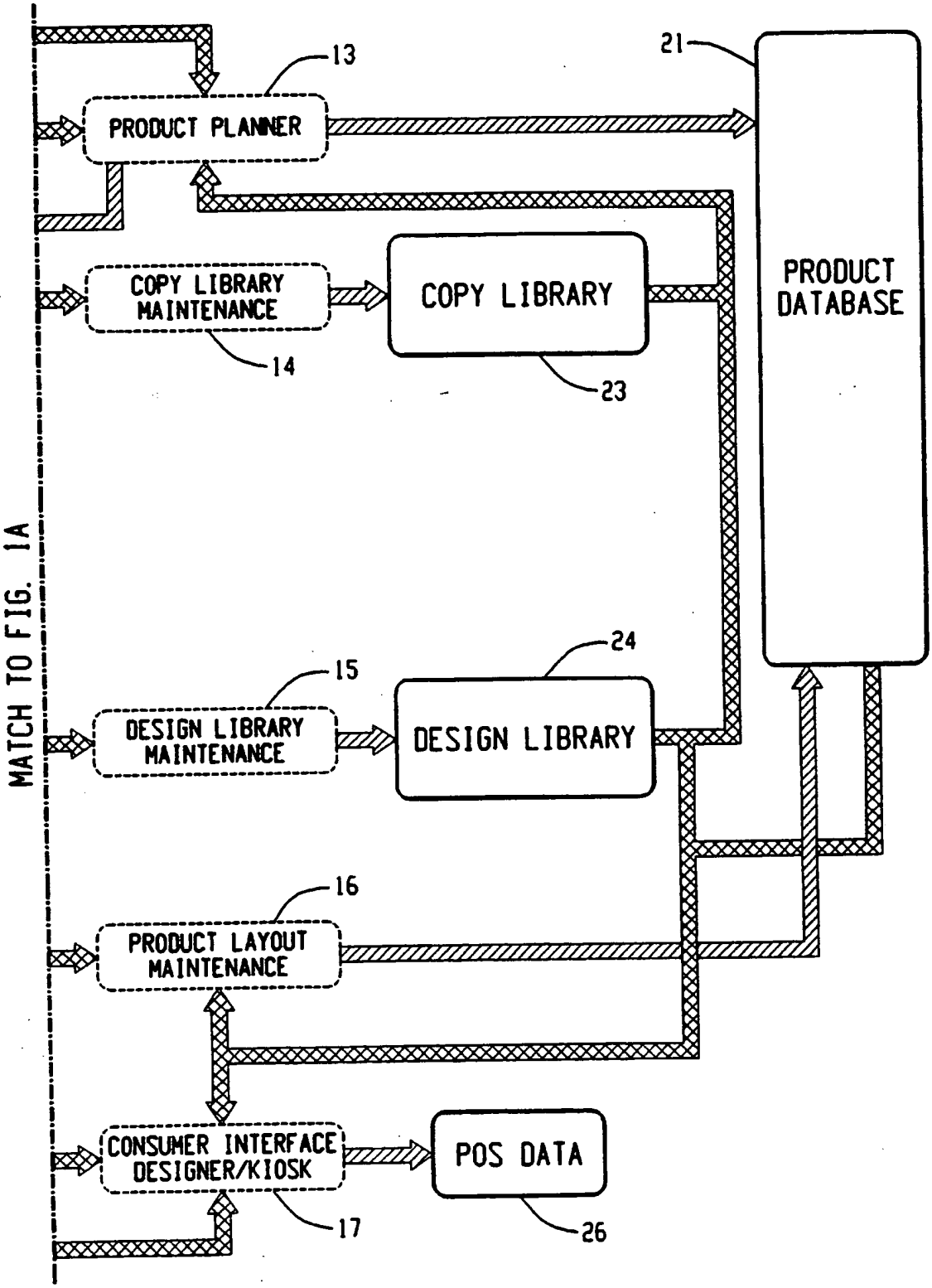

MATCH TO FIG. 1A
PRODUCT PLANNER
13
21
PRODUCT DATABASE
COPY LIBRARY MAINTENANCE
14
COPY LIBRARY
23
DESIGN LIBRARY MAINTENANCE
15
DESIGN LIBRARY
24
PRODUCT LAYOUT MAINTENANCE
16
CONSUMER INTERFACE DESIGNER/KIOSK
17
POS DATA
26

FIG. 1B

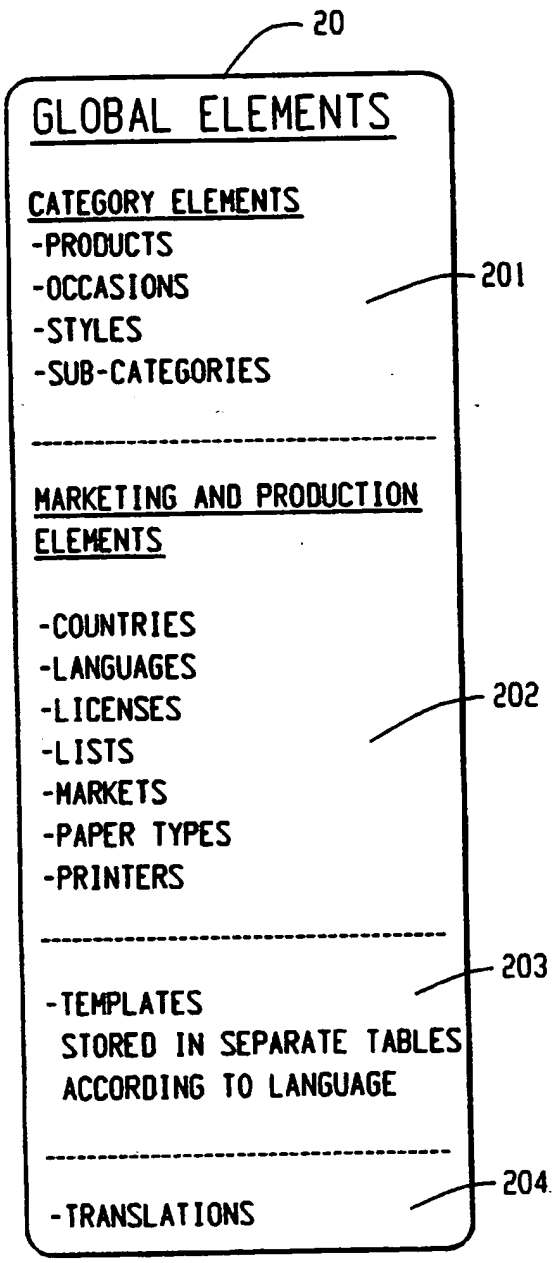

20
GLOBAL ELEMENTS
CATEGORY ELEMENTS
-PRODUCTS
-OCCASIONS
-STYLES
-SUB-CATEGORIES
201
MARKETING AND PRODUCTION ELEMENTS
-COUNTRIES
-LANGUAGES
-LICENSES
-LISTS
-MARKETS
-PAPER TYPES
-PRINTERS
202
-TEMPLATES
STORED IN SEPARATE TABLES ACCORDING TO LANGUAGE
203
-TRANSLATIONS
204

FIG. 2

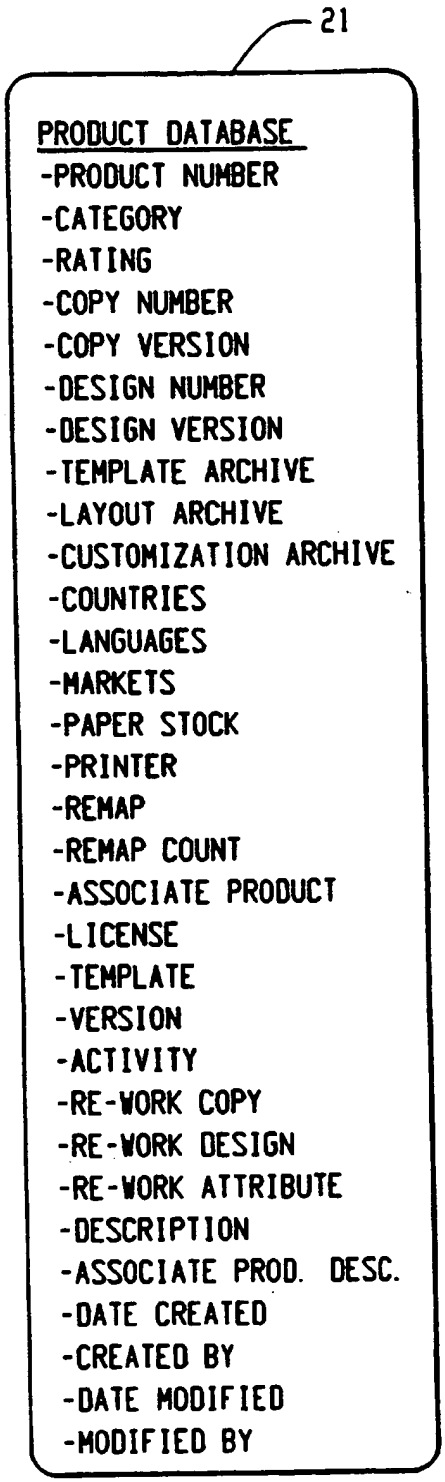

21
PRODUCT DATABASE
-PRODUCT NUMBER
-CATEGORY
-RATING
-COPY NUMBER
-COPY VERSION
-DESIGN NUMBER
-DESIGN VERSION
-TEMPLATE ARCHIVE
-LAYOUT ARCHIVE
-CUSTOMIZATION ARCHIVE
-COUNTRIES
-LANGUAGES
-MARKETS
-PAPER STOCK
-PRINTER
-REMAP
-REMAP COUNT
-ASSOCIATE PRODUCT
-LICENSE
-TEMPLATE
-VERSION
-ACTIVITY
-RE-WORK COPY
-RE-WORK DESIGN
-RE-WORK ATTRIBUTE
-DESCRIPTION
-ASSOCIATE PROD. DESC.
-DATE CREATED
-CREATED BY
-DATE MODIFIED
-MODIFIED BY

FIG. 3

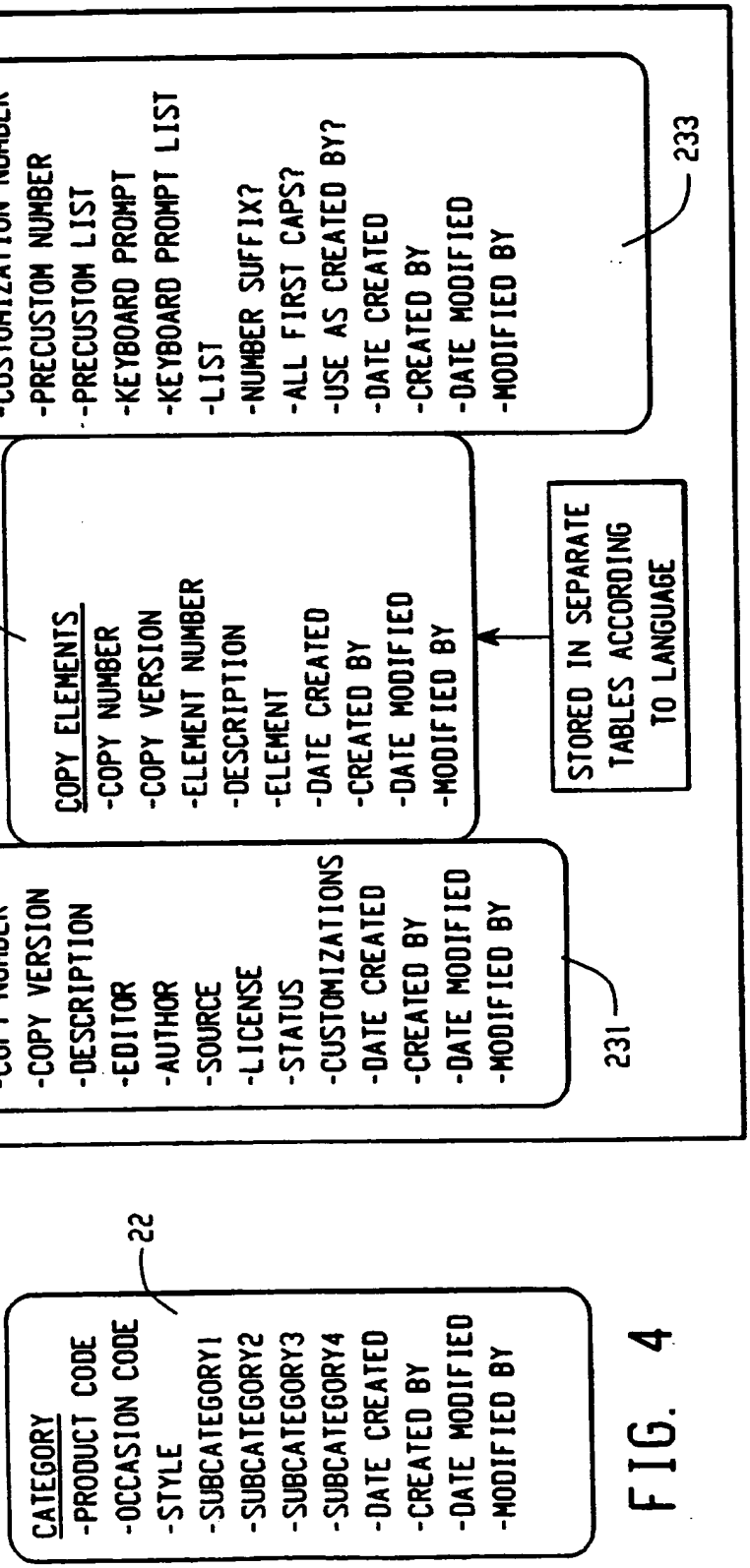

CATEGORY
-PRODUCT CODE
-OCCASION CODE
-STYLE
-SUBCATEGORY1
-SUBCATEGORY2
-SUBCATEGORY3
-SUBCATEGORY4
-DATE CREATED
-CREATED BY
-DATE MODIFIED
-MODIFIED BY
22

FIG. 4

23
COPY LIBRARY
COPY
-COPY NUMBER
-COPY VERSION
-DESCRIPTION
-EDITOR
-AUTHOR
-SOURCE
-LICENSE
-STATUS
-CUSTOMIZATIONS
-DATE CREATED
-CREATED BY
-DATE MODIFIED
-MODIFIED BY
231
COPY ELEMENTS
-COPY NUMBER
-COPY VERSION
-ELEMENT NUMBER
-DESCRIPTION
-ELEMENT
-DATE CREATED
-CREATED BY
-DATE MODIFIED
-MODIFIED BY
232
CUSTOMIZATION FORMATS
-CUSTOMIZATION NUMBER
-PRECUSTOM NUMBER
-PRECUSTOM LIST
-KEYBOARD PROMPT
-KEYBOARD PROMPT LIST
-LIST
-NUMBER SUFFIX?
-ALL FIRST CAPS?
-USE AS CREATED BY?
-DATE CREATED
-CREATED BY
-DATE MODIFIED
-MODIFIED BY
233
STORED IN SEPARATE TABLES ACCORDING TO LANGUAGE

FIG. 5

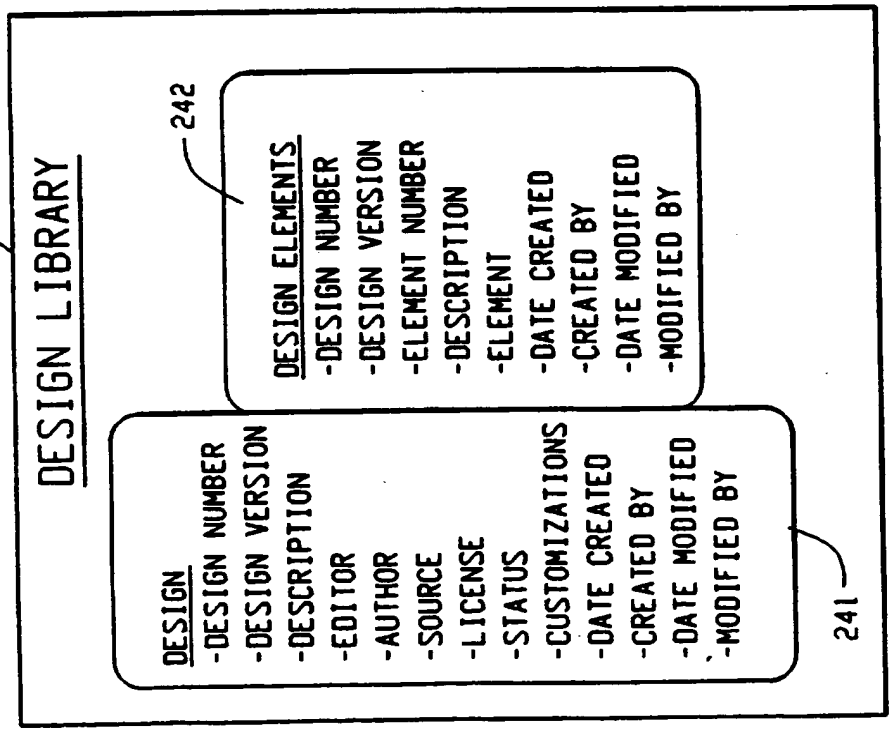


FIG. 6

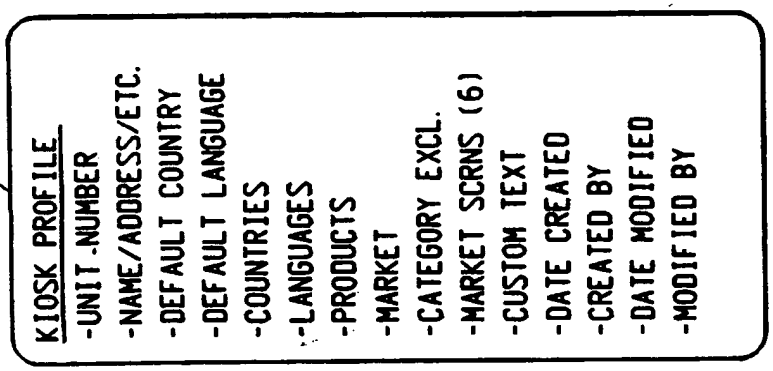


FIG. 7

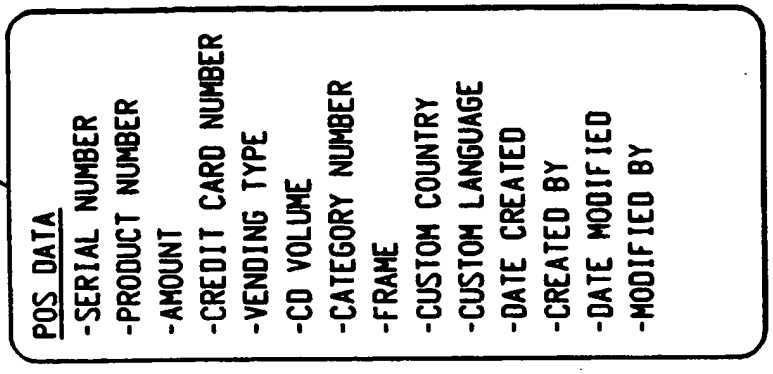


FIG. 8

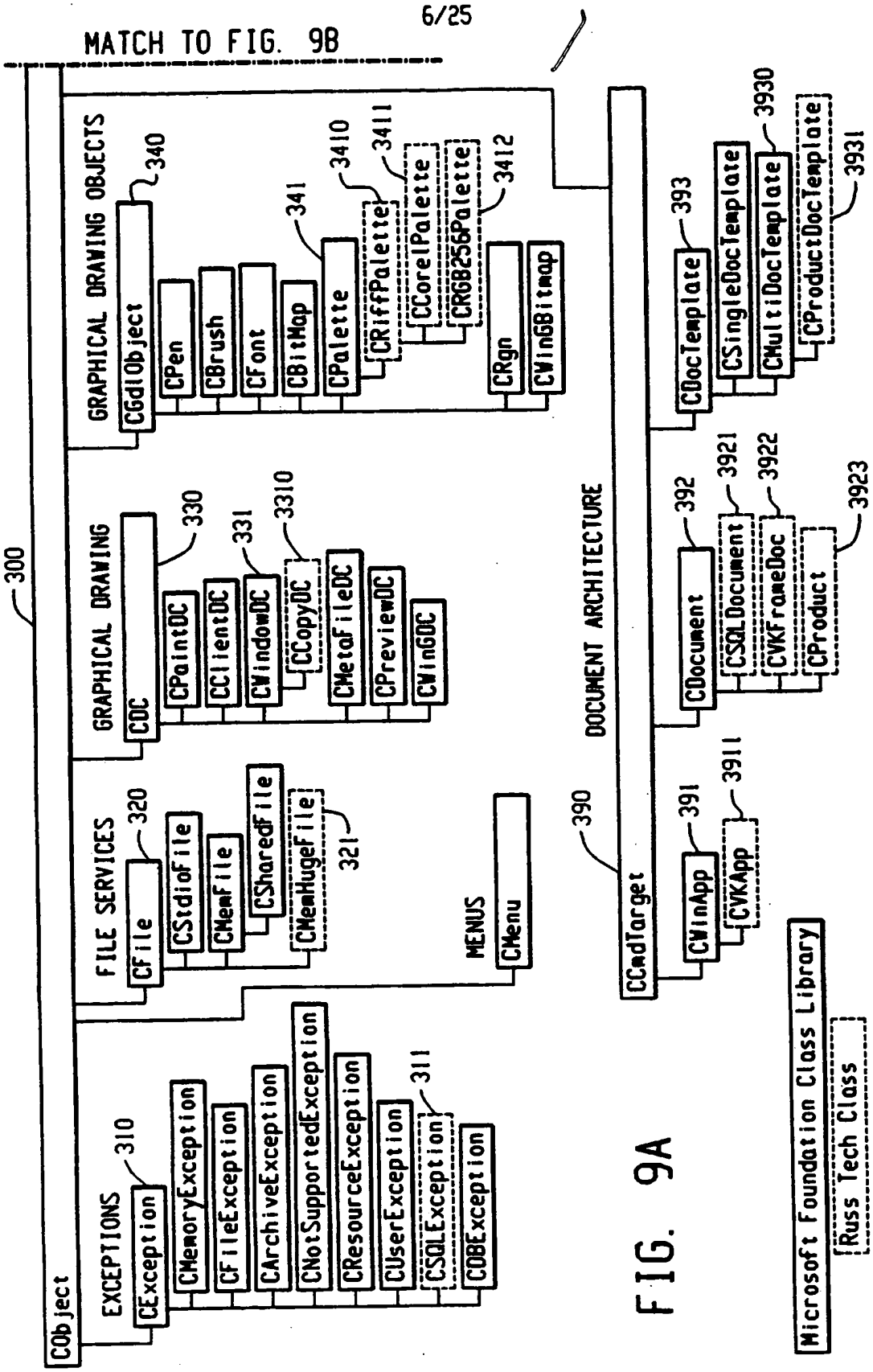

MATCH TO FIG. 9B
300
CObject
EXCEPTIONS
310
CException
CMemoryException
CFileException
CArchiveException
CNotSupportedException
CResourceException
CUserException
CSQLException
311
COBException
FILE SERVICES
320
CFile
CStdioFile
CMemFile
CSharedFile
CMemHugeFile
321
MENUS
CMenu
GRAPHICAL DRAWING
CDC
330
CPaintDC
CClientDC
CWindowDC
331
CCopyDC
3310
CMetaFileDC
CPreviewDC
CWinGDC
GRAPHICAL DRAWING OBJECTS
CGdiObject
340
CPen
CBrush
CFont
CBitMap
341
CPalette
CRiffPalette
3410
CCorelPalette
3411
CRGB256Palette
3412
CRgn
CWinGBitmap
DOCUMENT ARCHITECTURE
CCmdTarget
390
CWinApp
391
CVKApp
3911
CDocument
392
CDocTemplate
393
CSingleDocTemplate
3930
CMultiDocTemplate
CProductDocTemplate
3931
CSQLDocument
3921
CVKFrameDoc
3922
CProduct
3923
Microsoft Foundation Class Library
Russ Tech Class

FIG. 9A

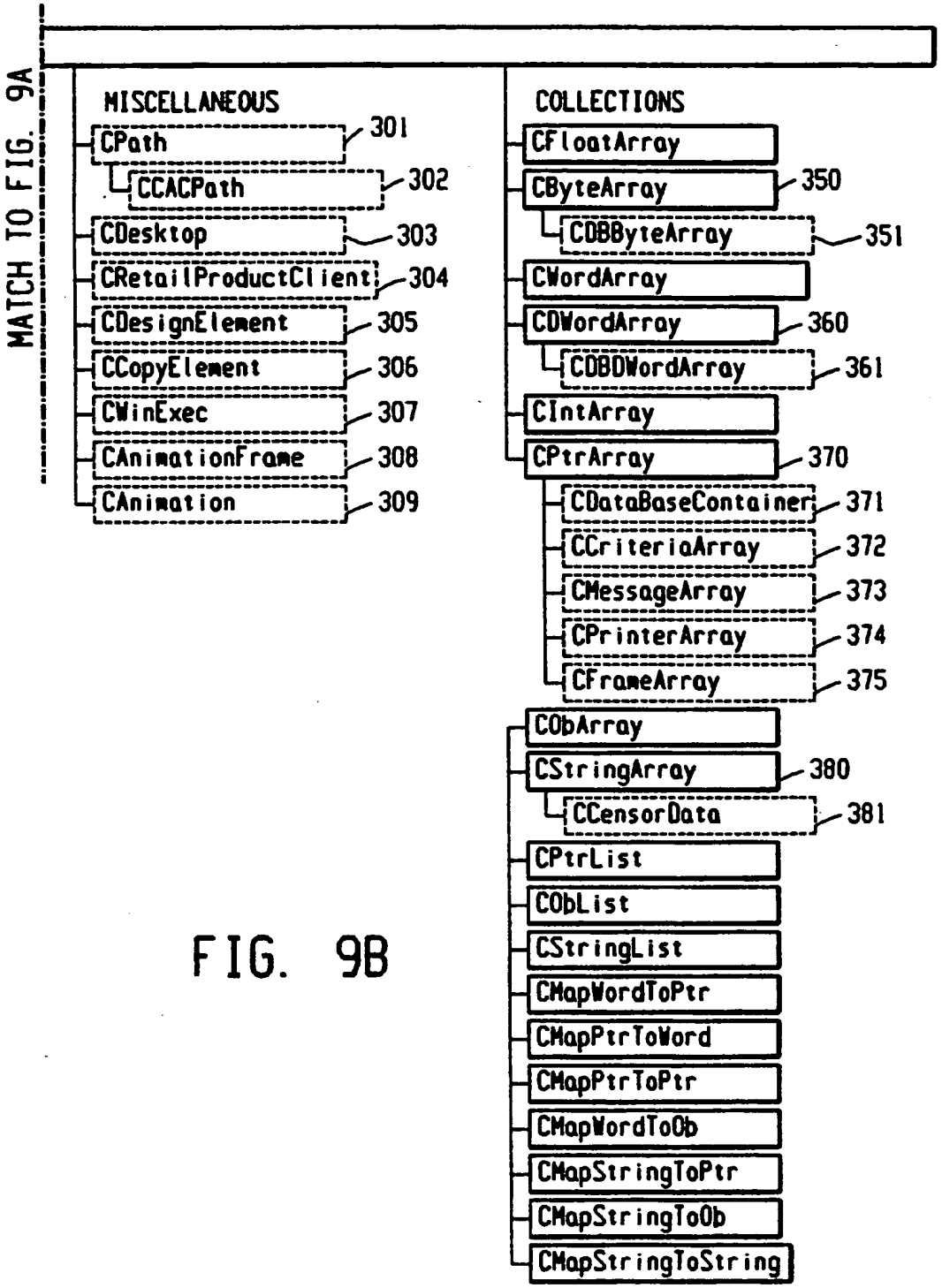

MATCH TO FIG. 9A
MISCELLANEOUS
CPath
301
CCACPath
302
CDesktop
303
CRetailProductClient
304
CDesignElement
305
CCopyElement
306
CWinExec
307
CAnimationFrame
308
CAnimation
309
COLLECTIONS
CFloatArray
CByteArray
350
CDBByteArray
351
CWordArray
CDWordArray
360
CDBDWordArray
361
CIntArray
CPtrArray
370
CDataBaseContainer
371
CCriteriaArray
372
CMessageArray
373
CPrinterArray
374
CFrameArray
375
CObArray
CStringArray
380
CCensorData
381
CPtrList
CObList
CStringList
CMapWordToPtr
CMapPtrToWord
CMapPtrToPtr
CMapWordToOb
CMapStringToPtr
CMapStringToOb
CMapStringToString

FIG. 9B

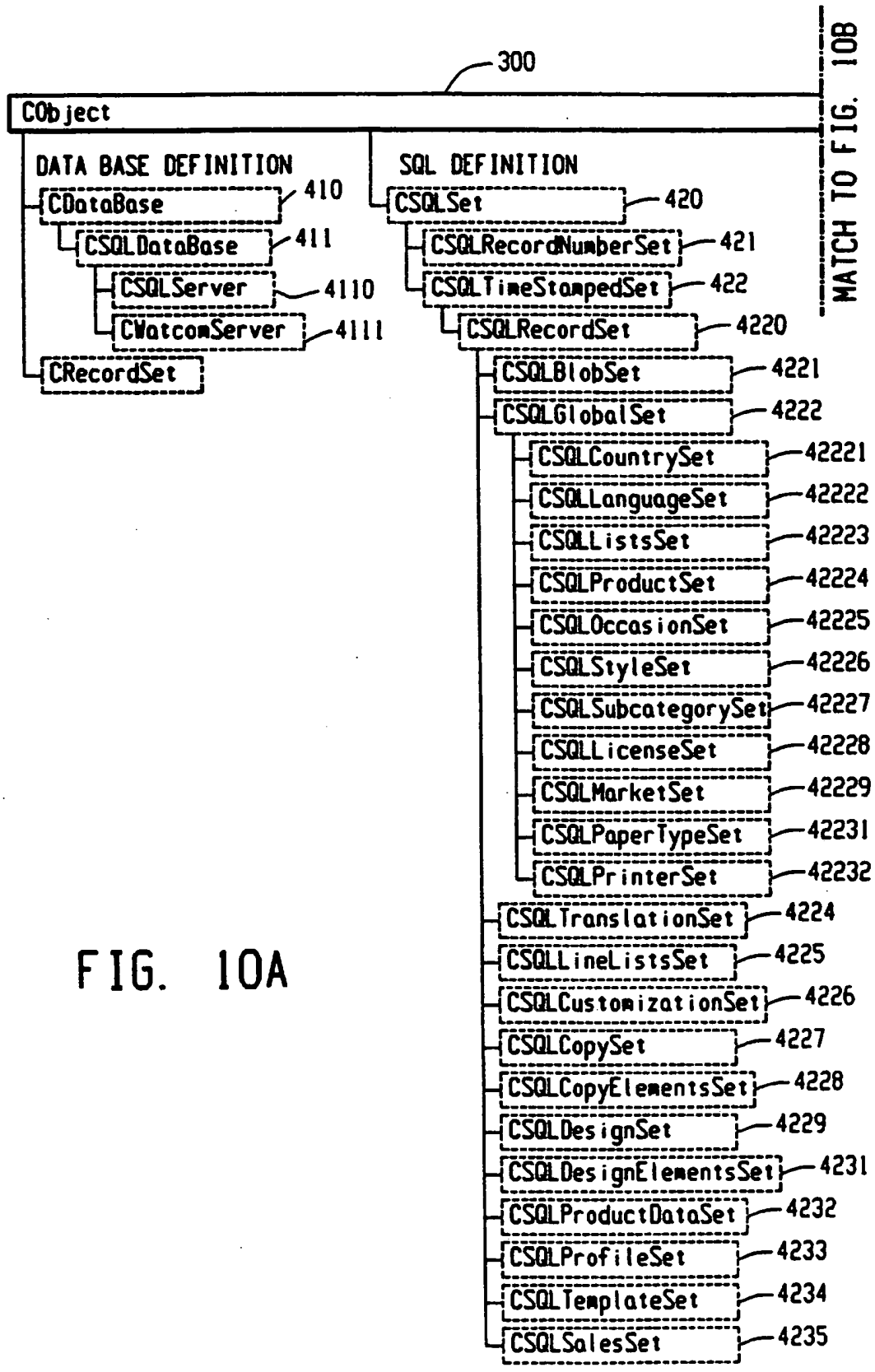

300
CObject
DATA BASE DEFINITION
SQL DEFINITION
CDataBase
410
CSQLDataBase
411
CSQLServer
4110
CWatcomServer
4111
CRecordSet
CSQLSet
420
CSQLRecordNumberSet
421
CSQLTimeStampedSet
422
CSQLRecordSet
4220
CSQLBlobSet
4221
CSQLGlobalSet
4222
CSQLCountrySet
42221
CSQLLanguageSet
42222
CSQLListsSet
42223
CSQLProductSet
42224
CSQLOccasionSet
42225
CSQLStyleSet
42226
CSQLSubcategorySet
42227
CSQLLicenseSet
42228
CSQLMarketSet
42229
CSQLPaperTypeSet
42231
CSQLPrinterSet
42232
CSQLTranslationSet
4224
CSQLLineListsSet
4225
CSQLCustomizationSet
4226
CSQLCopySet
4227
CSQLCopyElementsSet
4228
CSQLDesignSet
4229
CSQLDesignElementsSet
4231
CSQLProductDataSet
4232
CSQLProfileSet
4233
CSQLTemplateSet
4234
CSQLSalesSet
4235
MATCH TO FIG. 10B

FIG. 10A

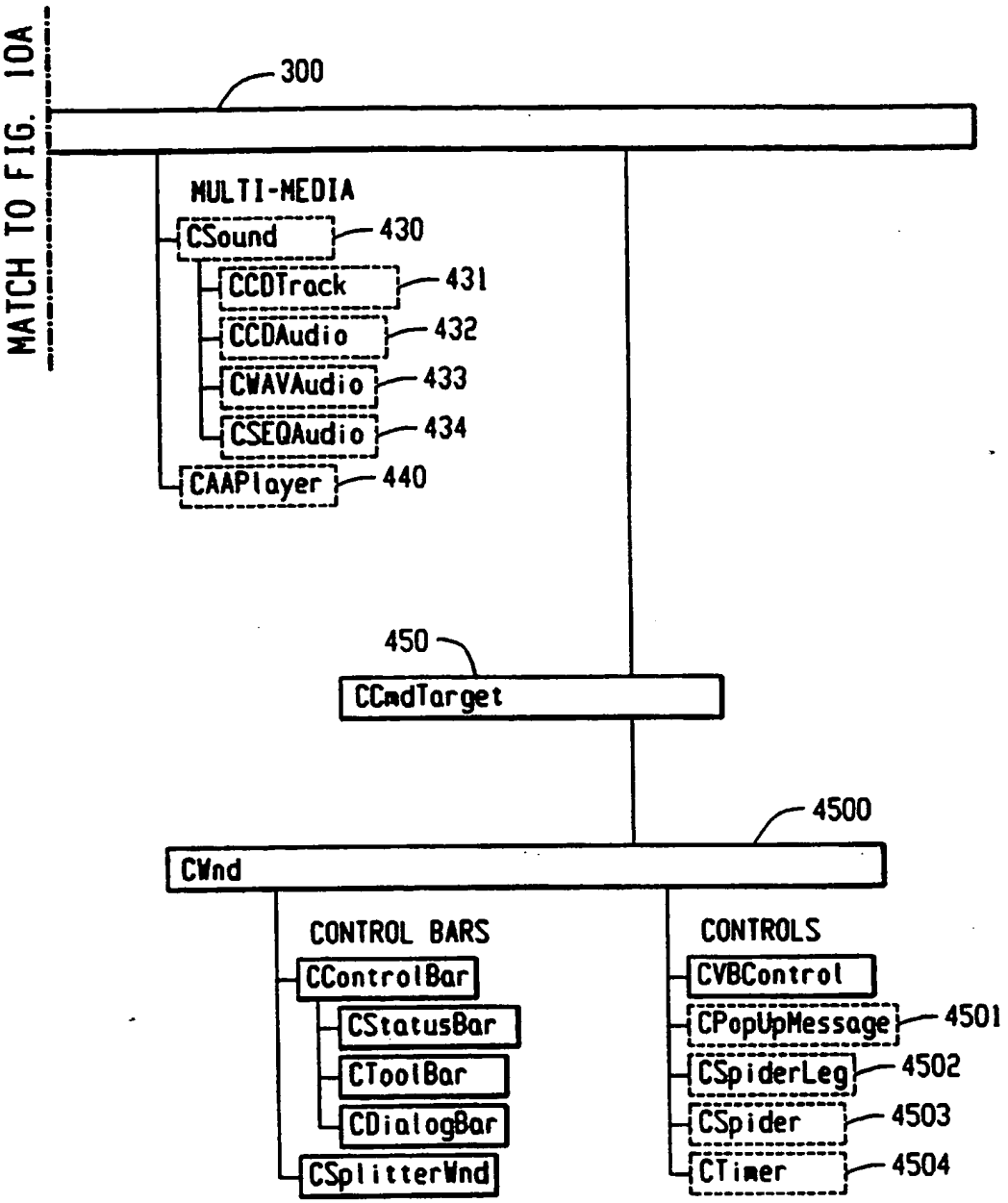

MATCH TO FIG. 10A
300
MULTI-MEDIA
CSound
430
CCDTrack
431
CCDAudio
432
CWAVAudio
433
CSEQAudio
434
CAAPlayer
440
450
CCmdTarget
4500
CWnd
CONTROL BARS
CControlBar
CStatusBar
CToolBar
CDialogBar
CSplitterWnd
CONTROLS
CVBControl
CPopUpMessage
4501
CSpiderLeg
4502
CSpider
4503
CTimer
4504

FIG. 10B

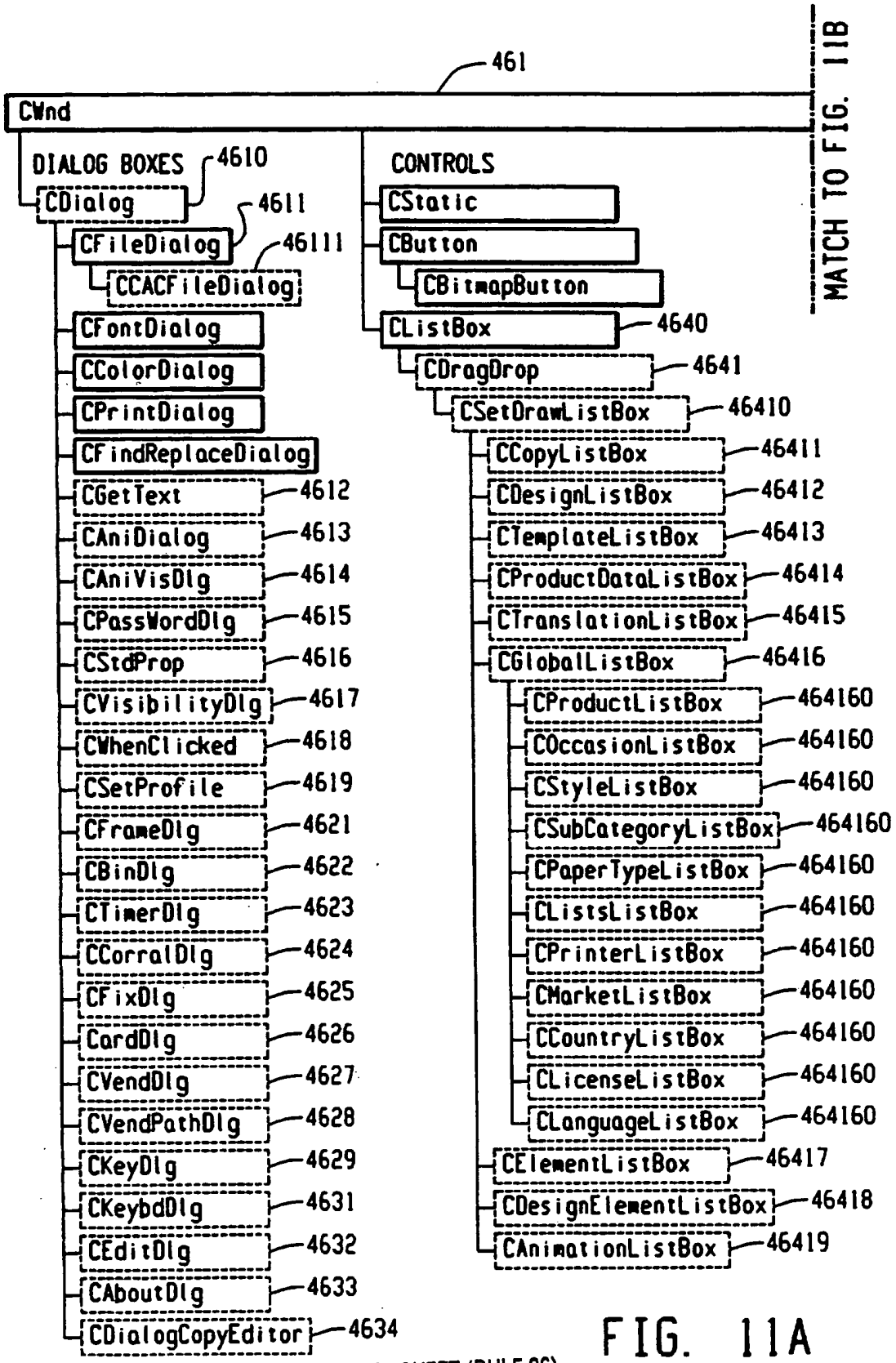

461
CWnd
MATCH TO FIG. 11B
DIALOG BOXES
4610
CDialog
4611
CFileDialog
46111
CCACFileDialog
CFontDialog
CColorDialog
CPrintDialog
CFindReplaceDialog
CGetText
4612
CAniDialog
4613
CAniVisDlg
4614
CPassWordDlg
4615
CStdProp
4616
CVisibilityDlg
4617
CWhenClicked
4618
CSetProfile
4619
CFrameDlg
4621
CBinDlg
4622
CTimerDlg
4623
CCorralDlg
4624
CFixDlg
4625
CardDlg
4626
CVendDlg
4627
CVendPathDlg
4628
CKeyDlg
4629
CKeybdDlg
4631
CEditDlg
4632
CAboutDlg
4633
CDialogCopyEditor
4634
CONTROLS
CStatic
CButton
CBitmapButton
CListBox
4640
CDragDrop
4641
CSetDrawListBox
46410
CCopyListBox
46411
CDesignListBox
46412
CTemplateListBox
46413
CProductDataListBox
46414
CTranslationListBox
46415
CGlobalListBox
46416
CProductListBox
464160
COccasionListBox
464160
CStyleListBox
464160
CSubCategoryListBox
464160
CPaperTypeListBox
464160
CListsListBox
464160
CPrinterListBox
464160
CMarketListBox
464160
CCountryListBox
464160
CLicenseListBox
464160
CLanguageListBox
464160
CElementListBox
46417
CDesignElementListBox
46418
CAnimationListBox
46419

FIG. 11A

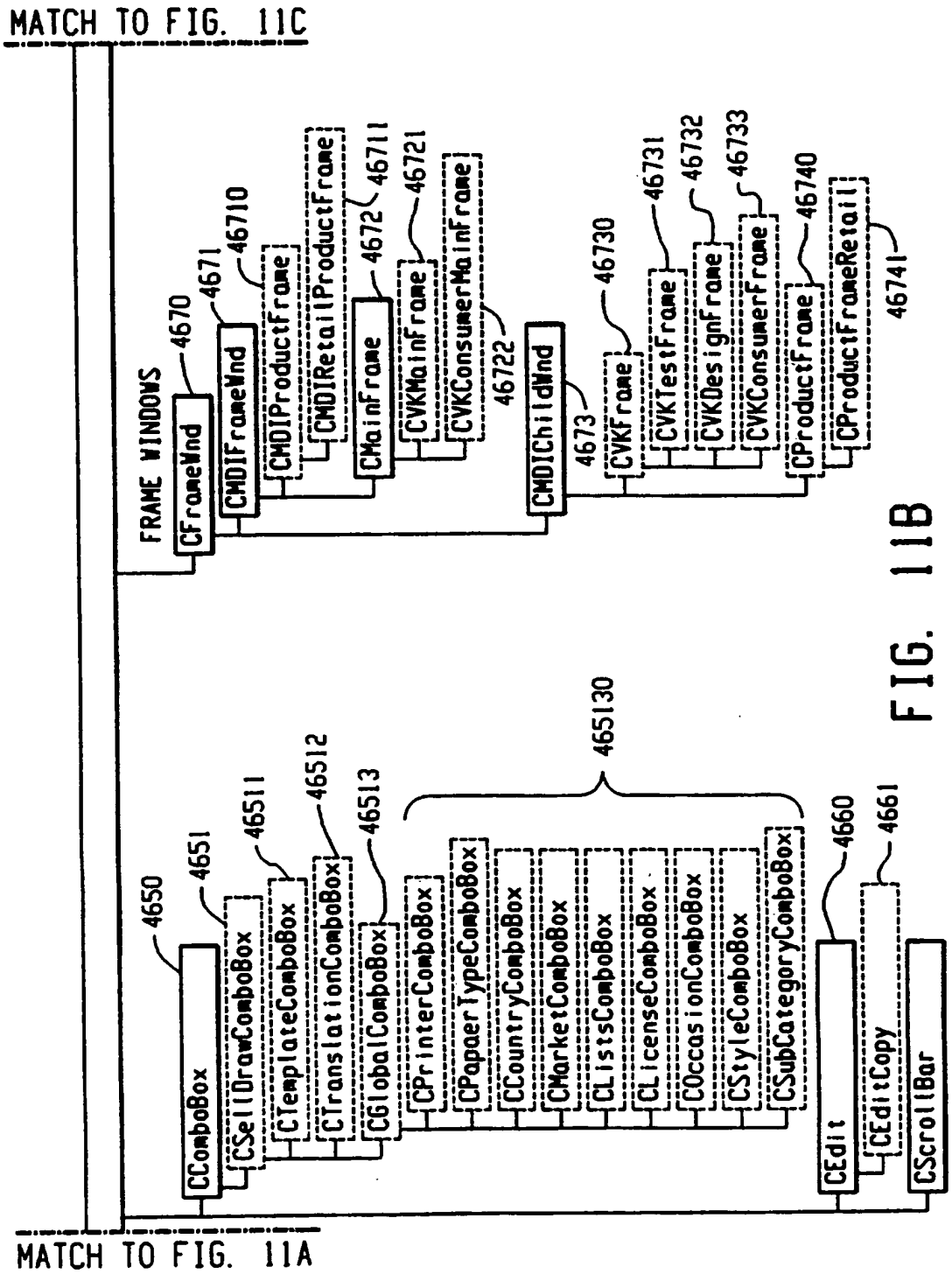

MATCH TO FIG. 11C
FRAME WINDOWS
CFrameWnd
4670
CMDIFrameWnd
4671
CMDIProductFrame
46710
CMDIRetailProductFrame
46711
CMainFrame
4672
CVKMainFrame
46721
CVKConsumerMainFrame
46722
CMDIChildWnd
4673
CVKFrame
46730
CVKTestFrame
46731
CVKDesignFrame
46732
CVKConsumerFrame
46733
CProductFrame
46740
CProductFrameRetail
46741
CComboBox
4650
CSellDrawComboBox
4651
CTemplateComboBox
46511
CTranslationComboBox
46512
CGlobalComboBox
46513
CPrinterComboBox
CPaperTypeComboBox
CCountryComboBox
CMarketComboBox
CListsComboBox
CLicenseComboBox
COccasionComboBox
CStyleComboBox
CSubCategoryComboBox
465130
CEdit
4660
CEditCopy
4661
CScrollBar
MATCH TO FIG. 11A

FIG. 11B

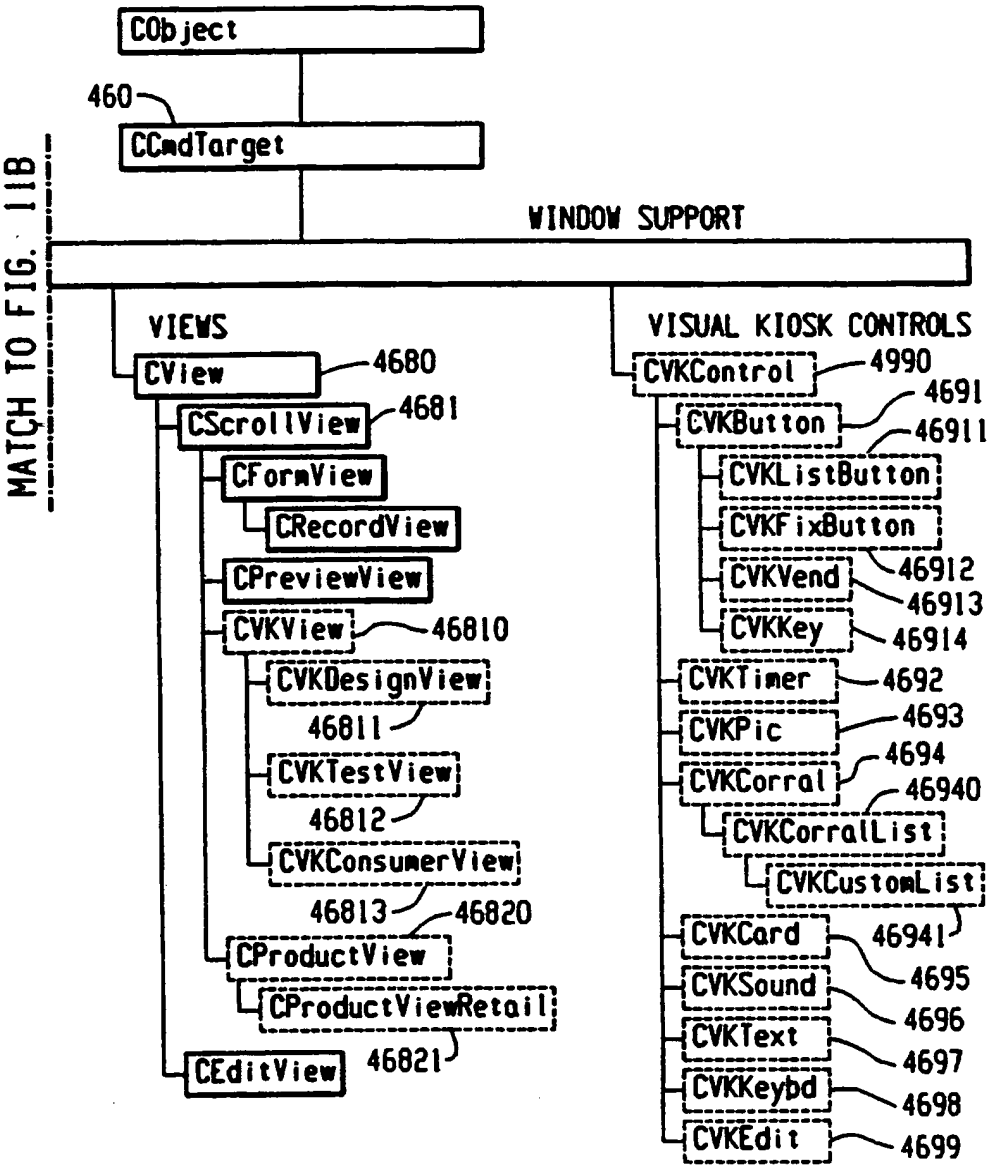

CObject
460
CCmdTarget
WINDOW SUPPORT
MATCH TO FIG. 11B
VIEWS
CView
4680
CScrollView
4681
CFormView
CRecordView
CPreviewView
CVKView
46810
CVKDesignView
46811
CVKTestView
46812
CVKConsumerView
46813
46820
CProductView
CProductViewRetail
CEditView
46821
VISUAL KIOSK CONTROLS
CVKControl
4990
CVKButton
4691
46911
CVKListButton
CVKFixButton
CVKVend
46912
46913
CVKKey
46914
CVKTimer
4692
CVKPic
4693
CVKCorral
4694
46940
CVKCorralList
CVKCustomList
CVKCard
46941
CVKSound
4695
4696
CVKText
4697
CVKKeybd
4698
CVKEdit
4699

FIG. 11C

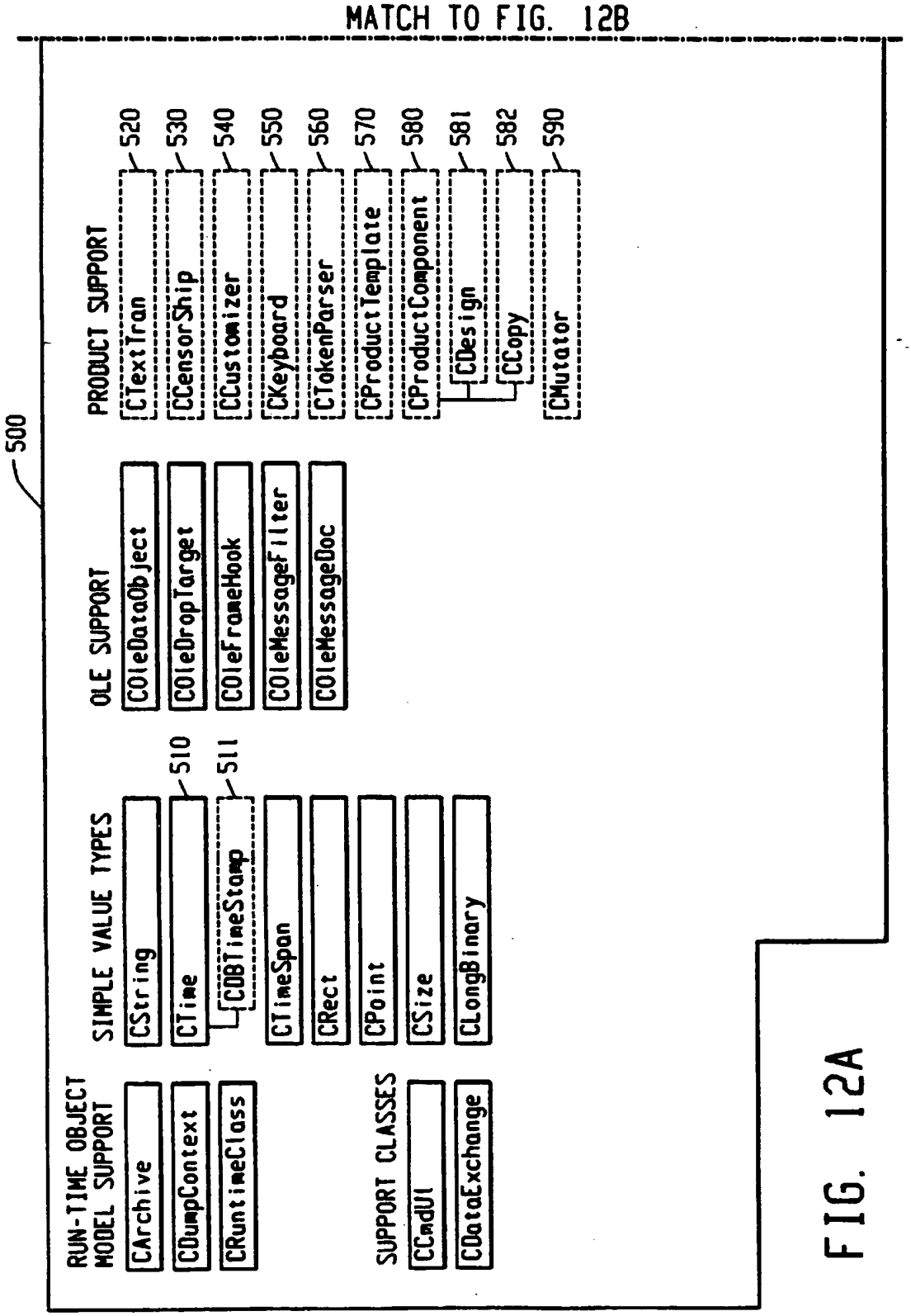

MATCH TO FIG. 12B
500
RUN-TIME OBJECT MODEL SUPPORT
CArchive
CDumpContext
CRuntimeClass
SIMPLE VALUE TYPES
CString
CTime
510
CDBTimeStamp
511
CTimeSpan
CRect
CPoint
CSize
CLongBinary
SUPPORT CLASSES
CCmdUI
CDataExchange
OLE SUPPORT
COleDataObject
COleDropTarget
COleFrameHook
COleMessageFilter
COleMessageDoc
PRODUCT SUPPORT
CTextTran
520
CCensorShip
530
CCustomizer
540
CKeyboard
550
CTokenParser
560
CProductTemplate
570
CProductComponent
580
CDesign
581
CCopy
582
CMutator
590

FIG. 12A

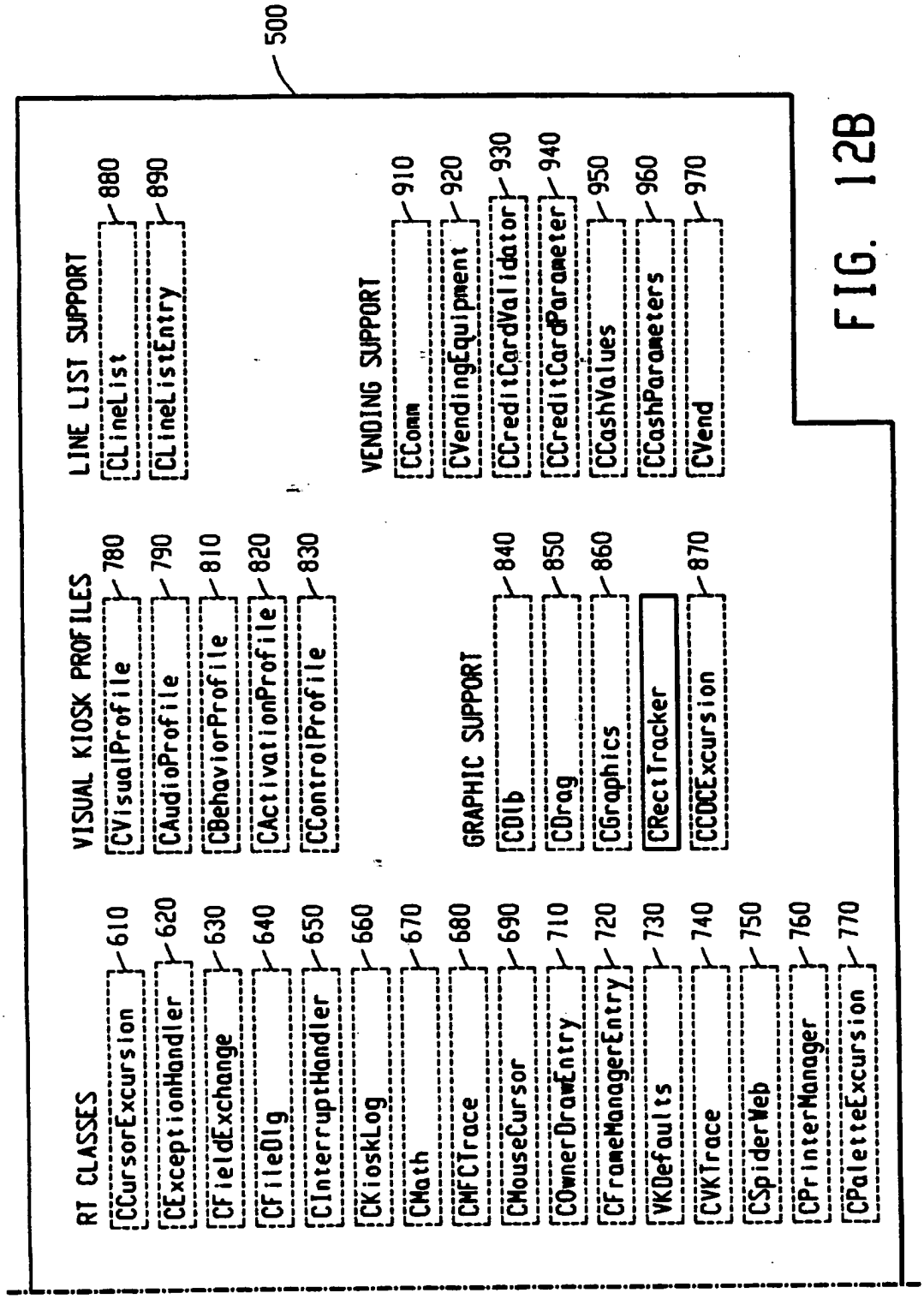


FIG. 12B

MATCH TO FIG. 12A

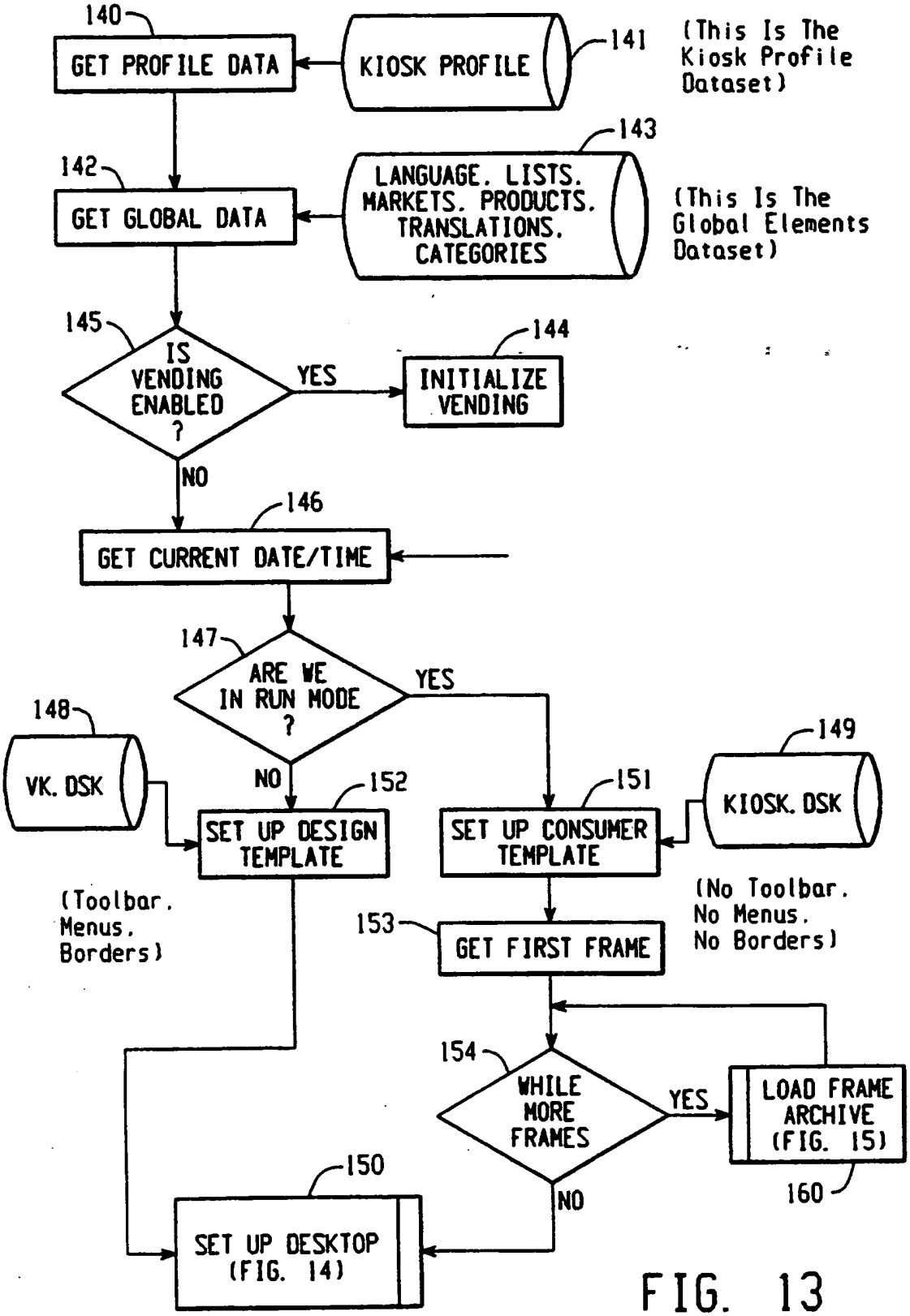
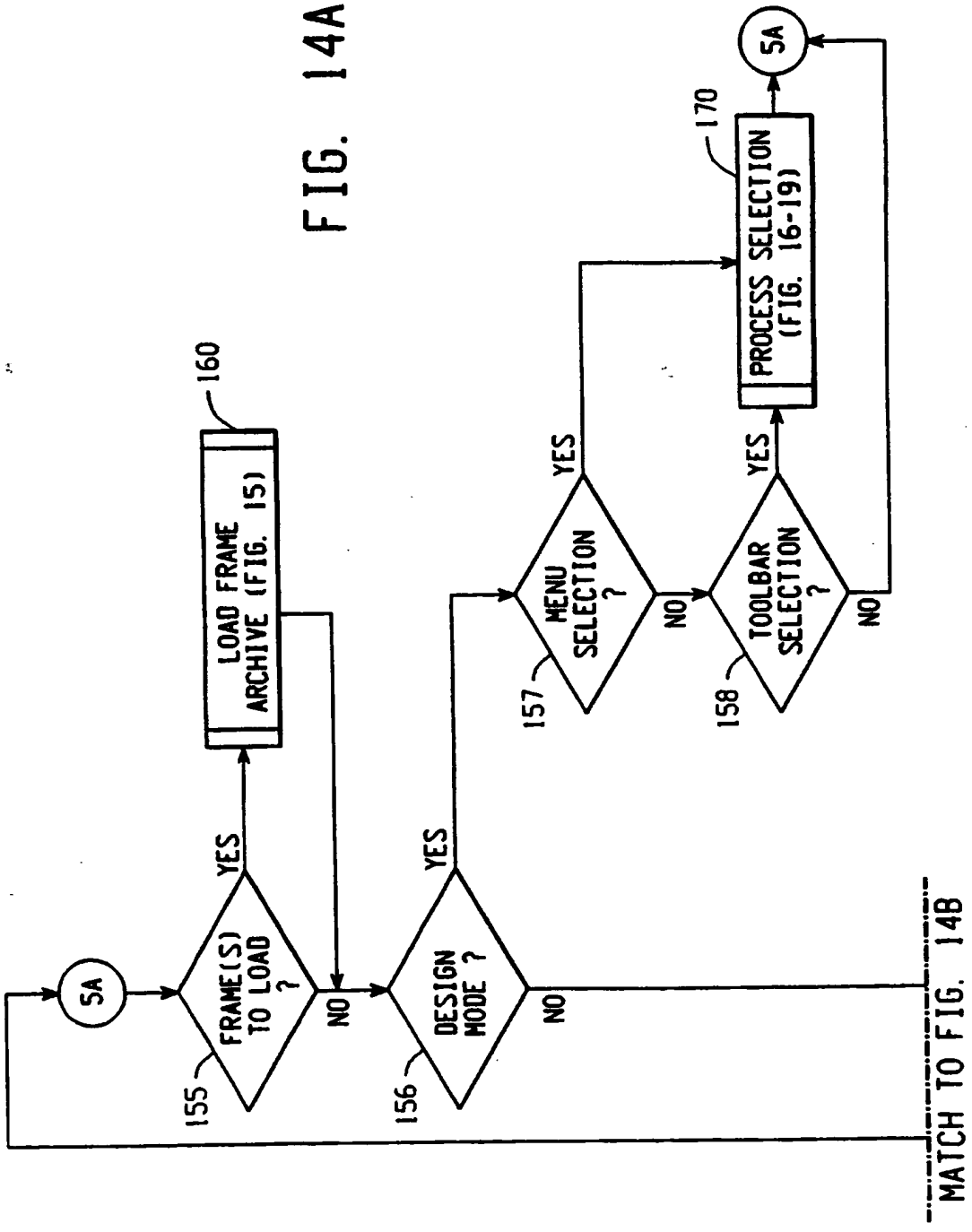

140
GET PROFILE DATA
KIOSK PROFILE
141
(This Is The Kiosk Profile Dataset)
143
142
GET GLOBAL DATA
LANGUAGE. LISTS. MARKETS. PRODUCTS. TRANSLATIONS. CATEGORIES
(This Is The Global Elements Dataset)
145
IS VENDING ENABLED ?
YES
144
INITIALIZE VENDING
NO
146
GET CURRENT DATE/TIME
147
ARE WE IN RUN MODE ?
YES
148
VK. DSK
NO
152
SET UP DESIGN TEMPLATE
151
SET UP CONSUMER TEMPLATE
149
KIOSK. DSK
(Toolbar. Menus. Borders)
(No Toolbar. No Menus. No Borders)
153
GET FIRST FRAME
154
WHILE MORE FRAMES
YES
LOAD FRAME ARCHIVE (FIG. 15)
160
NO
150
SET UP DESKTOP (FIG. 14)

FIG. 13

FIG. 14A


5A
FRAME(S) TO LOAD ?
155
YES
NO
LOAD FRAME ARCHIVE (FIG. 15)
160
DESIGN MODE ?
156
YES
NO
MENU SELECTION ?
157
YES
NO
TOOLBAR SELECTION ?
158
YES
NO
PROCESS SELECTION (FIG. 16-19)
170
5A
MATCH TO FIG. 14B

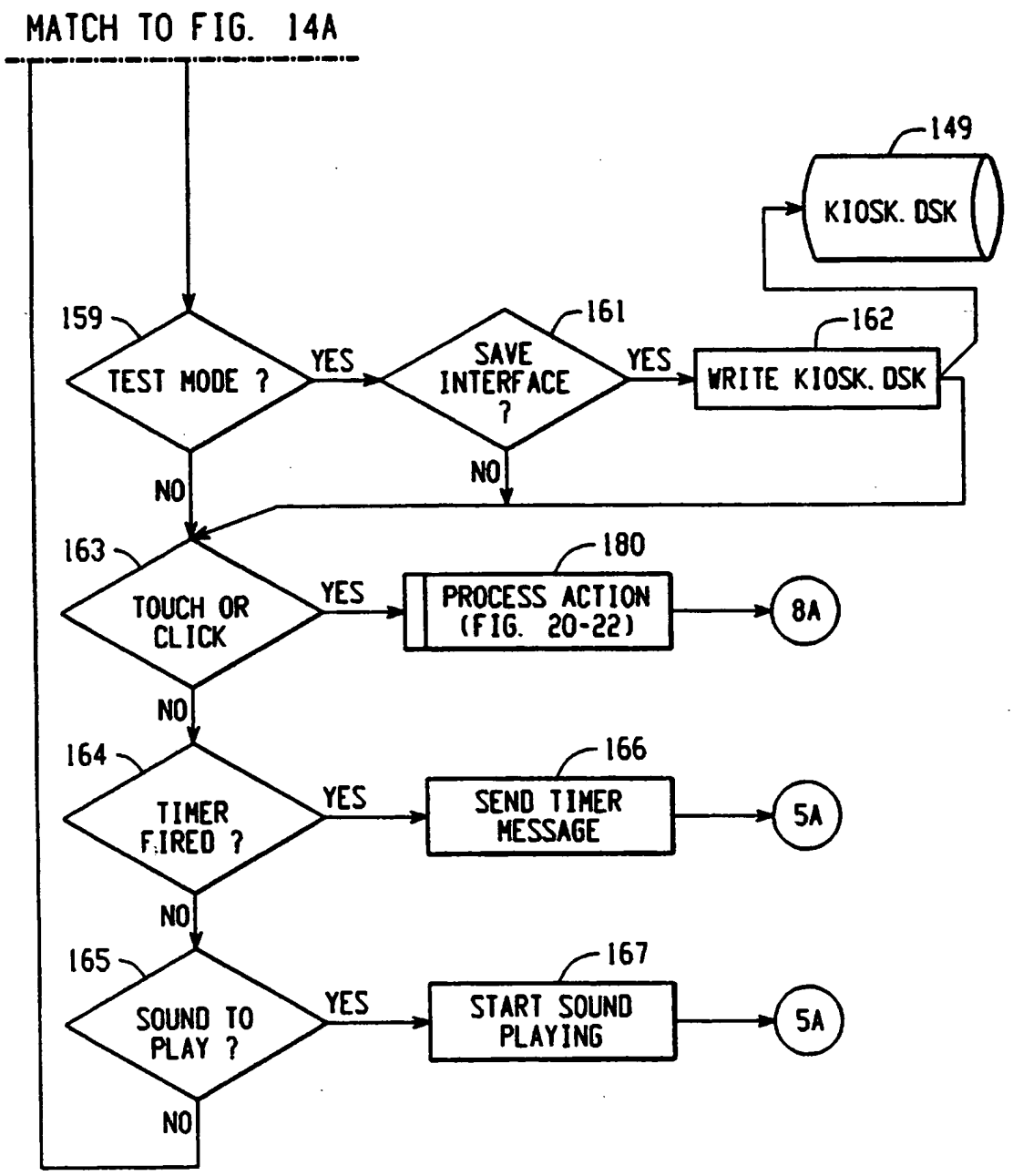

MATCH TO FIG. 14A
149
KIOSK. DSK
159
TEST MODE ?
YES
161
SAVE INTERFACE ?
YES
162
WRITE KIOSK. DSK
NO
NO
163
TOUCH OR CLICK
YES
180
PROCESS ACTION (FIG. 20-22)
8A
NO
164
TIMER FIRED ?
YES
166
SEND TIMER MESSAGE
5A
NO
165
SOUND TO PLAY ?
YES
167
START SOUND PLAYING
5A
NO

FIG. 14B

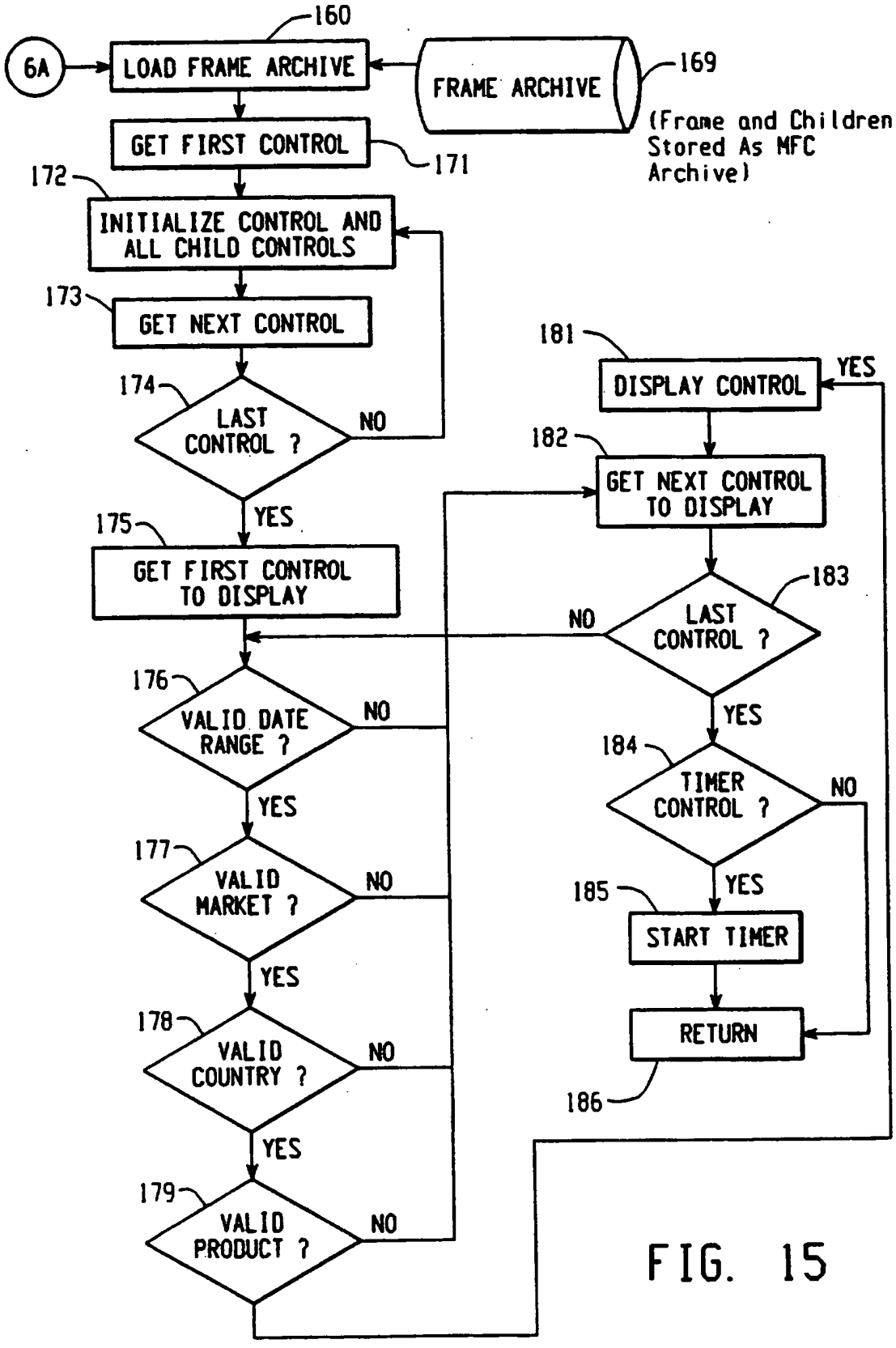

6A
160
LOAD FRAME ARCHIVE
FRAME ARCHIVE
169
(Frame and Children Stored As MFC Archive)
GET FIRST CONTROL
171
172
INITIALIZE CONTROL AND ALL CHILD CONTROLS
173
GET NEXT CONTROL
174
LAST CONTROL ?
NO
YES
175
GET FIRST CONTROL TO DISPLAY
176
VALID DATE RANGE ?
NO
YES
177
VALID MARKET ?
NO
YES
178
VALID COUNTRY ?
NO
YES
179
VALID PRODUCT ?
NO
181
DISPLAY CONTROL
YES
182
GET NEXT CONTROL TO DISPLAY
183
LAST CONTROL ?
NO
YES
184
TIMER CONTROL ?
NO
YES
185
START TIMER
RETURN
186

FIG. 15

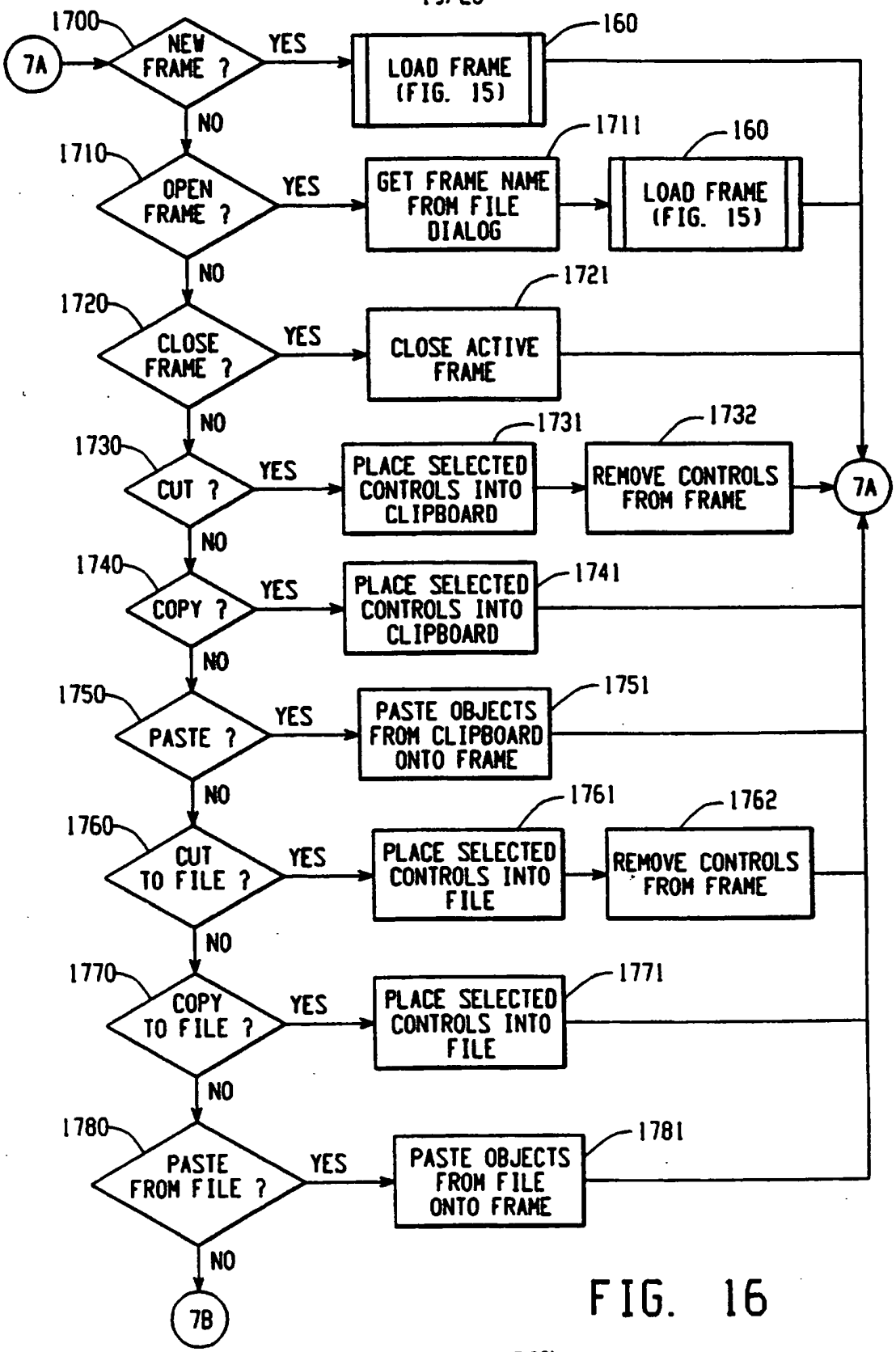

7A
1700
NEW FRAME ?
YES
LOAD FRAME (FIG. 15)
160
NO
1710
OPEN FRAME ?
YES
GET FRAME NAME FROM FILE DIALOG
1711
LOAD FRAME (FIG. 15)
160
NO
1720
CLOSE FRAME ?
YES
CLOSE ACTIVE FRAME
1721
NO
1730
CUT ?
YES
PLACE SELECTED CONTROLS INTO CLIPBOARD
1731
REMOVE CONTROLS FROM FRAME
1732
7A
NO
1740
COPY ?
YES
PLACE SELECTED CONTROLS INTO CLIPBOARD
1741
NO
1750
PASTE ?
YES
PASTE OBJECTS FROM CLIPBOARD ONTO FRAME
1751
NO
1760
CUT TO FILE ?
YES
PLACE SELECTED CONTROLS INTO FILE
1761
REMOVE CONTROLS FROM FRAME
1762
NO
1770
COPY TO FILE ?
YES
PLACE SELECTED CONTROLS INTO FILE
1771
NO
1780
PASTE FROM FILE ?
YES
PASTE OBJECTS FROM FILE ONTO FRAME
1781
NO
7B

FIG. 16

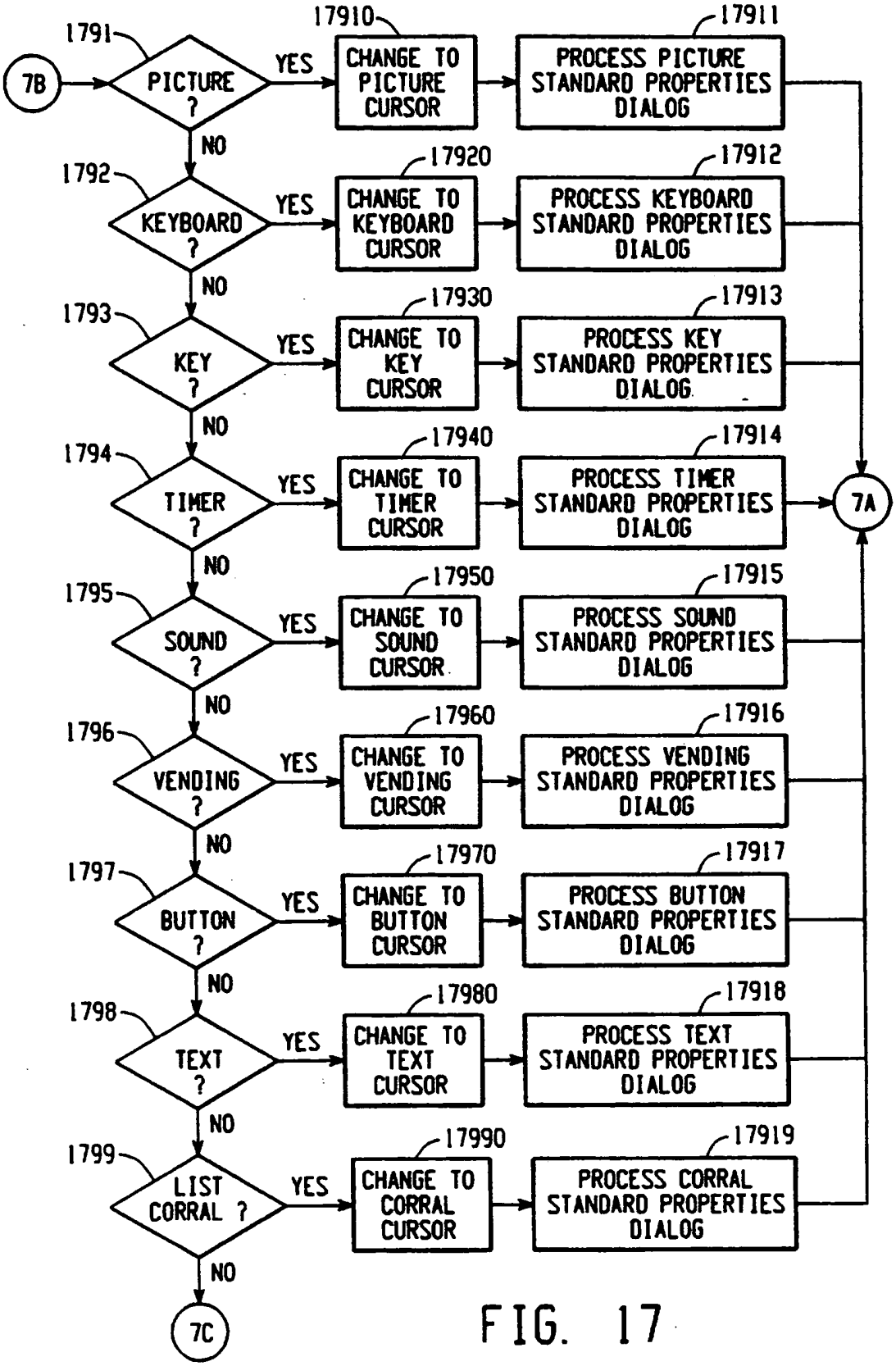

7B
1791
PICTURE ?
YES
17910
CHANGE TO PICTURE CURSOR
17911
PROCESS PICTURE STANDARD PROPERTIES DIALOG
NO
1792
KEYBOARD ?
YES
17920
CHANGE TO KEYBOARD CURSOR
17912
PROCESS KEYBOARD STANDARD PROPERTIES DIALOG
NO
1793
KEY ?
YES
17930
CHANGE TO KEY CURSOR
17913
PROCESS KEY STANDARD PROPERTIES DIALOG
NO
1794
TIMER ?
YES
17940
CHANGE TO TIMER CURSOR
17914
PROCESS TIMER STANDARD PROPERTIES DIALOG
7A
NO
1795
SOUND ?
YES
17950
CHANGE TO SOUND CURSOR
17915
PROCESS SOUND STANDARD PROPERTIES DIALOG
NO
1796
VENDING ?
YES
17960
CHANGE TO VENDING CURSOR
17916
PROCESS VENDING STANDARD PROPERTIES DIALOG
NO
1797
BUTTON ?
YES
17970
CHANGE TO BUTTON CURSOR
17917
PROCESS BUTTON STANDARD PROPERTIES DIALOG
NO
1798
TEXT ?
YES
17980
CHANGE TO TEXT CURSOR
17918
PROCESS TEXT STANDARD PROPERTIES DIALOG
NO
1799
LIST CORRAL ?
YES
17990
CHANGE TO CORRAL CURSOR
17919
PROCESS CORRAL STANDARD PROPERTIES DIALOG
NO
7C

FIG. 17

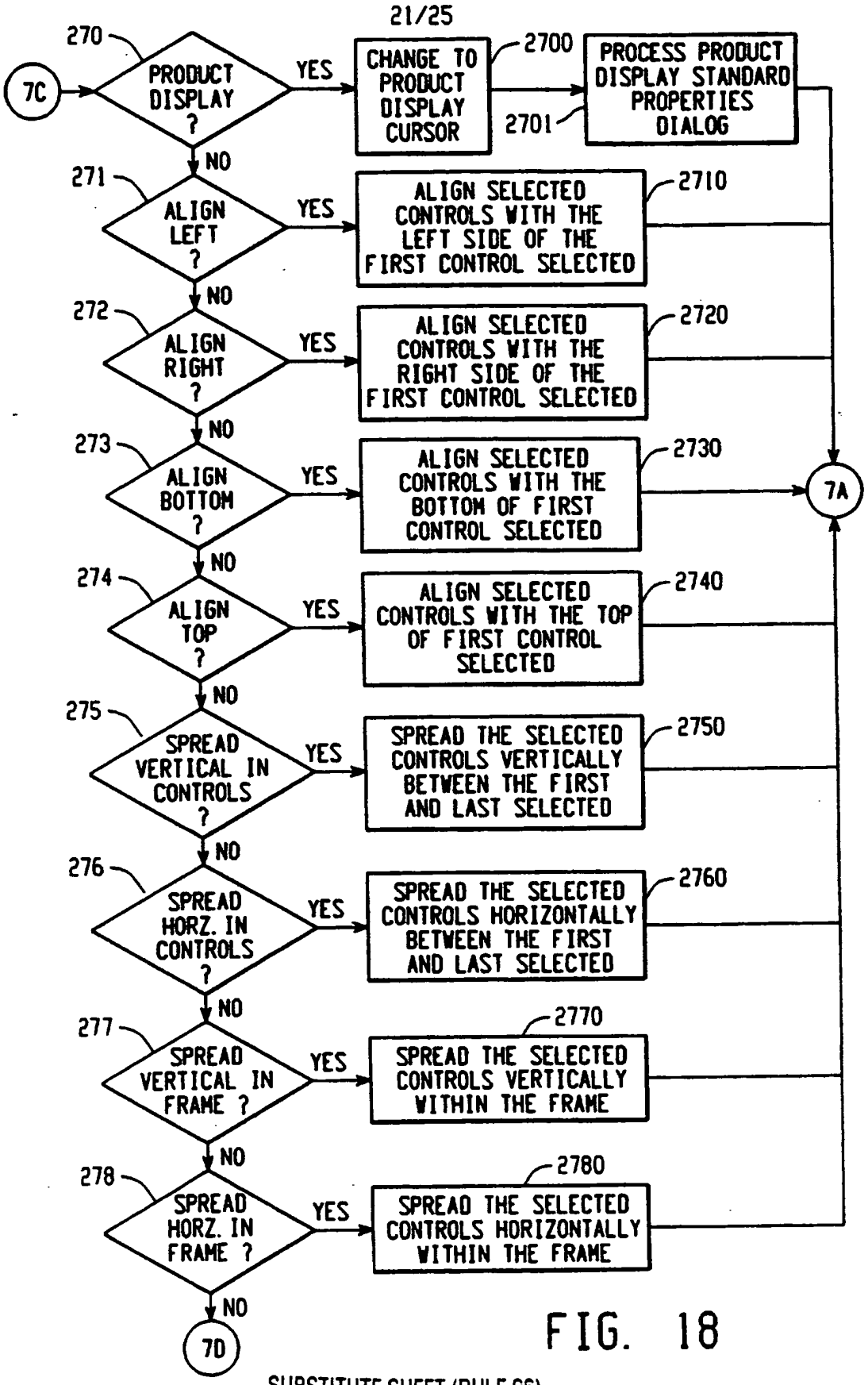

7C
270
PRODUCT DISPLAY ?
YES
CHANGE TO PRODUCT DISPLAY CURSOR
2700
PROCESS PRODUCT DISPLAY STANDARD PROPERTIES DIALOG
2701
NO
271
ALIGN LEFT ?
YES
ALIGN SELECTED CONTROLS WITH THE LEFT SIDE OF THE FIRST CONTROL SELECTED
2710
NO
272
ALIGN RIGHT ?
YES
ALIGN SELECTED CONTROLS WITH THE RIGHT SIDE OF THE FIRST CONTROL SELECTED
2720
NO
273
ALIGN BOTTOM ?
YES
ALIGN SELECTED CONTROLS WITH THE BOTTOM OF FIRST CONTROL SELECTED
2730
7A
NO
274
ALIGN TOP ?
YES
ALIGN SELECTED CONTROLS WITH THE TOP OF FIRST CONTROL SELECTED
2740
NO
275
SPREAD VERTICAL IN CONTROLS ?
YES
SPREAD THE SELECTED CONTROLS VERTICALLY BETWEEN THE FIRST AND LAST SELECTED
2750
NO
276
SPREAD HORZ. IN CONTROLS ?
YES
SPREAD THE SELECTED CONTROLS HORIZONTALLY BETWEEN THE FIRST AND LAST SELECTED
2760
NO
277
SPREAD VERTICAL IN FRAME ?
YES
SPREAD THE SELECTED CONTROLS VERTICALLY WITHIN THE FRAME
2770
NO
278
SPREAD HORZ. IN FRAME ?
YES
SPREAD THE SELECTED CONTROLS HORIZONTALLY WITHIN THE FRAME
2780
NO
7D

FIG. 18

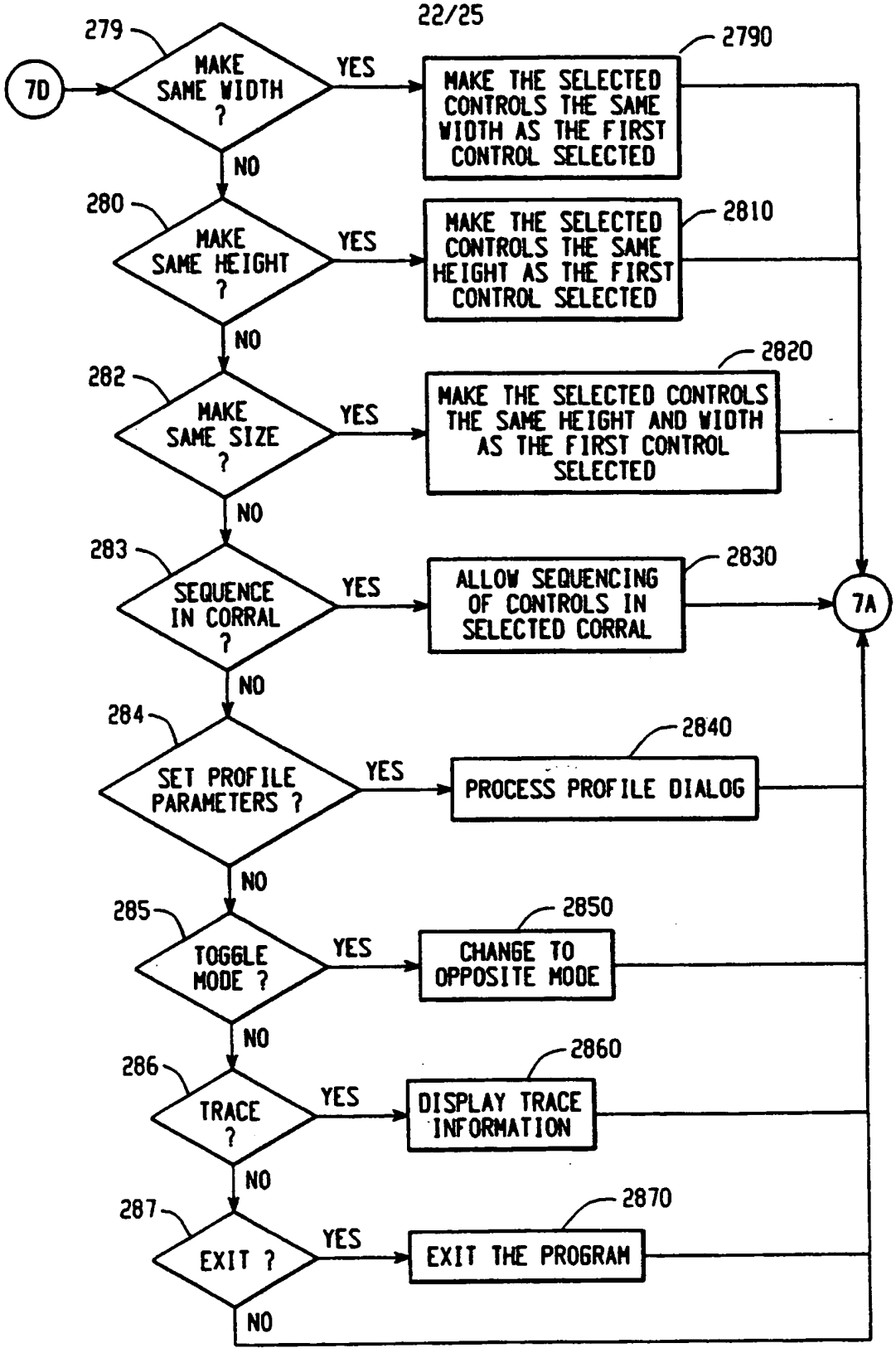

7D
279
MAKE SAME WIDTH ?
YES
2790
MAKE THE SELECTED CONTROLS THE SAME WIDTH AS THE FIRST CONTROL SELECTED
NO
280
MAKE SAME HEIGHT ?
YES
2810
MAKE THE SELECTED CONTROLS THE SAME HEIGHT AS THE FIRST CONTROL SELECTED
NO
282
MAKE SAME SIZE ?
YES
2820
MAKE THE SELECTED CONTROLS THE SAME HEIGHT AND WIDTH AS THE FIRST CONTROL SELECTED
NO
283
SEQUENCE IN CORRAL ?
YES
2830
ALLOW SEQUENCING OF CONTROLS IN SELECTED CORRAL
7A
NO
284
SET PROFILE PARAMETERS ?
YES
2840
PROCESS PROFILE DIALOG
NO
285
TOGGLE MODE ?
YES
2850
CHANGE TO OPPOSITE MODE
NO
286
TRACE ?
YES
2860
DISPLAY TRACE INFORMATION
NO
287
EXIT ?
YES
2870
EXIT THE PROGRAM
NO

FIG. 19

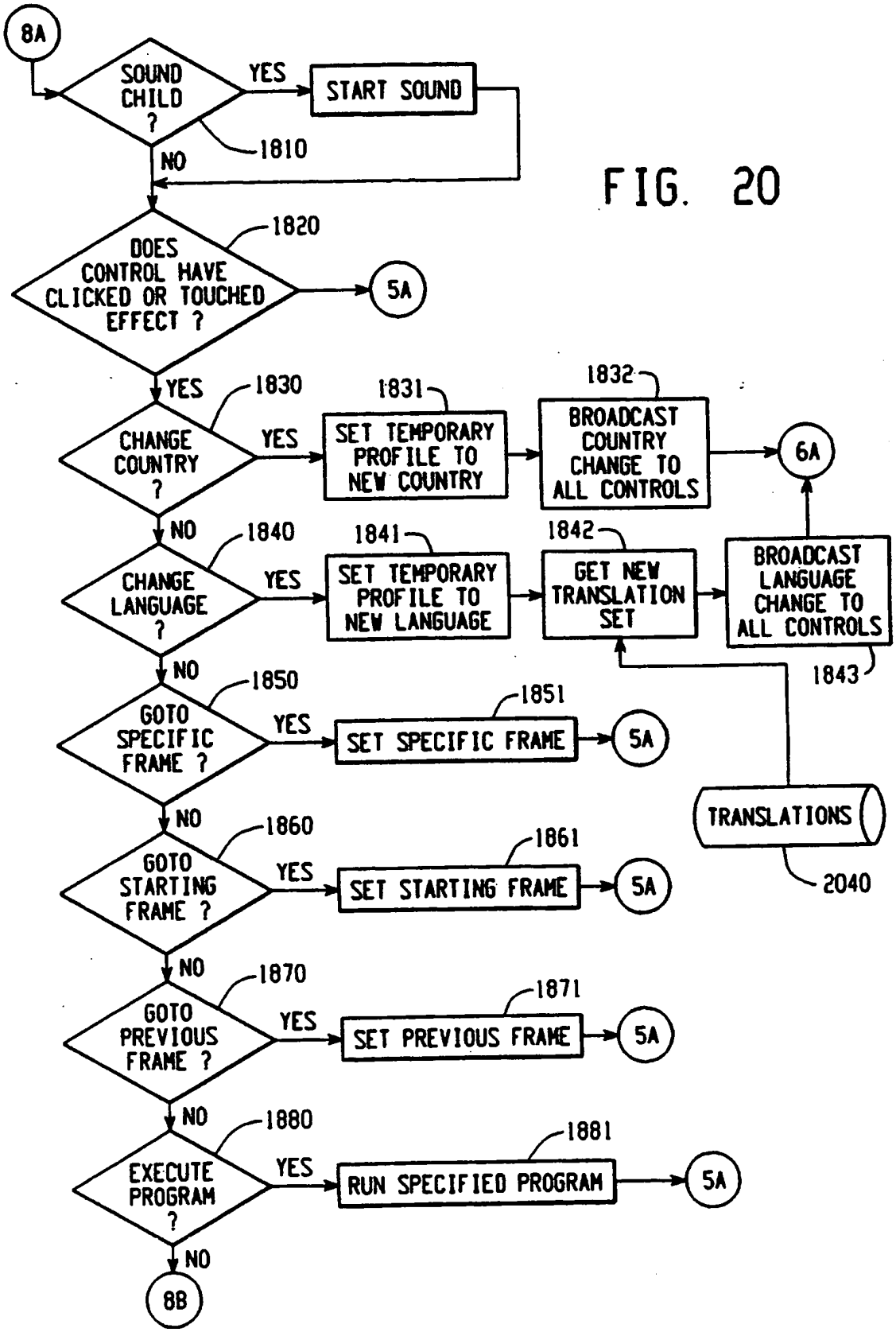

8A
SOUND CHILD ?
YES
START SOUND
1810
NO
1820
DOES CONTROL HAVE CLICKED OR TOUCHED EFFECT ?
5A
YES
1830
CHANGE COUNTRY ?
YES
1831
SET TEMPORARY PROFILE TO NEW COUNTRY
1832
BROADCAST COUNTRY CHANGE TO ALL CONTROLS
6A
NO
1840
CHANGE LANGUAGE ?
YES
1841
SET TEMPORARY PROFILE TO NEW LANGUAGE
1842
GET NEW TRANSLATION SET
BROADCAST LANGUAGE CHANGE TO ALL CONTROLS
1843
NO
1850
GOTO SPECIFIC FRAME ?
YES
1851
SET SPECIFIC FRAME
5A
TRANSLATIONS
2040
NO
1860
GOTO STARTING FRAME ?
YES
1861
SET STARTING FRAME
5A
NO
1870
GOTO PREVIOUS FRAME ?
YES
1871
SET PREVIOUS FRAME
5A
NO
1880
EXECUTE PROGRAM ?
YES
1881
RUN SPECIFIED PROGRAM
5A
NO
8B

FIG. 20

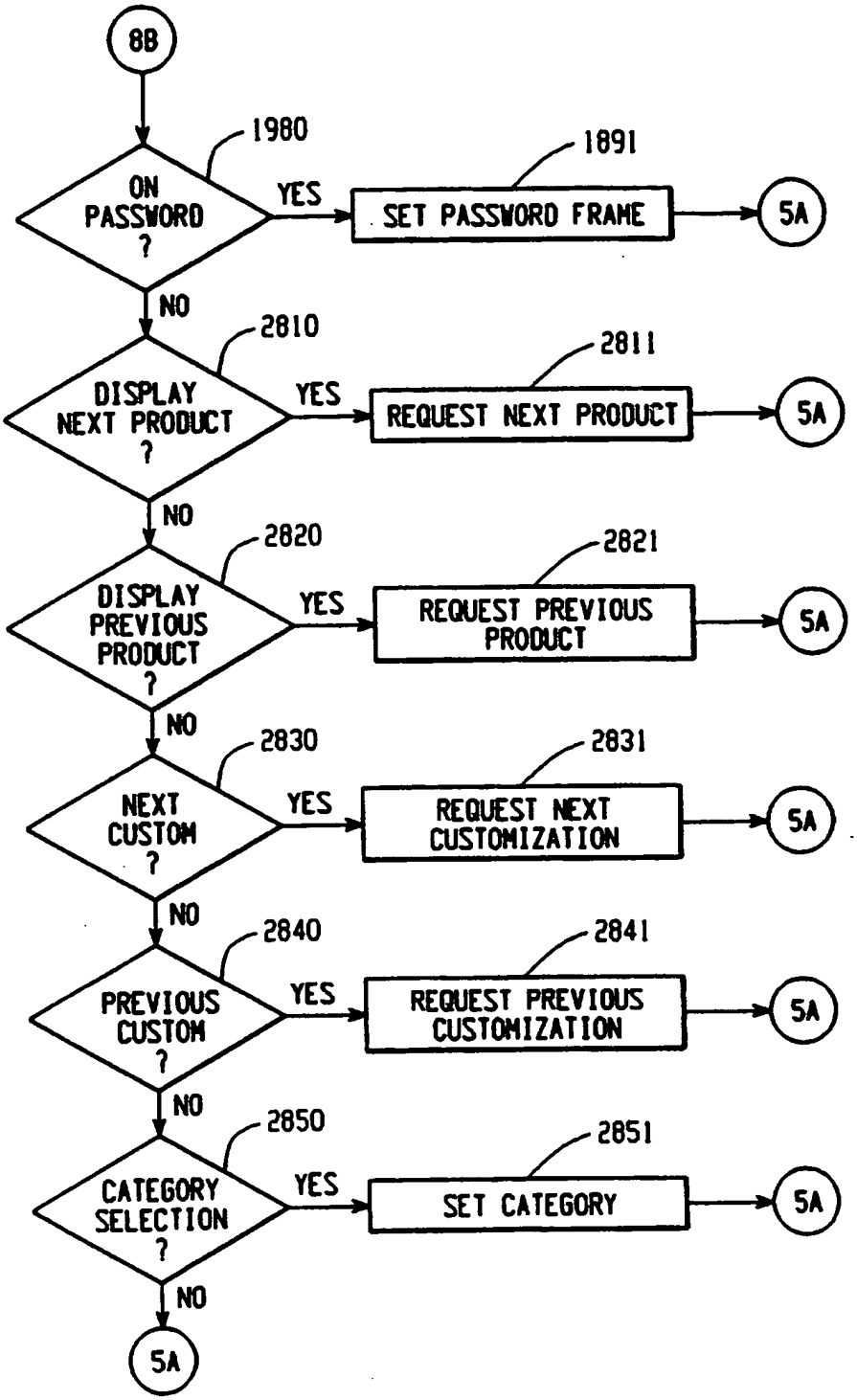

8B
1980
ON PASSWORD ?
YES
1891
SET PASSWORD FRAME
5A
NO
2810
DISPLAY NEXT PRODUCT ?
YES
2811
REQUEST NEXT PRODUCT
5A
NO
2820
DISPLAY PREVIOUS PRODUCT ?
YES
2821
REQUEST PREVIOUS PRODUCT
5A
NO
2830
NEXT CUSTOM ?
YES
2831
REQUEST NEXT CUSTOMIZATION
5A
NO
2840
PREVIOUS CUSTOM ?
YES
2841
REQUEST PREVIOUS CUSTOMIZATION
5A
NO
2850
CATEGORY SELECTION ?
YES
2851
SET CATEGORY
5A
NO
5A

FIG. 21

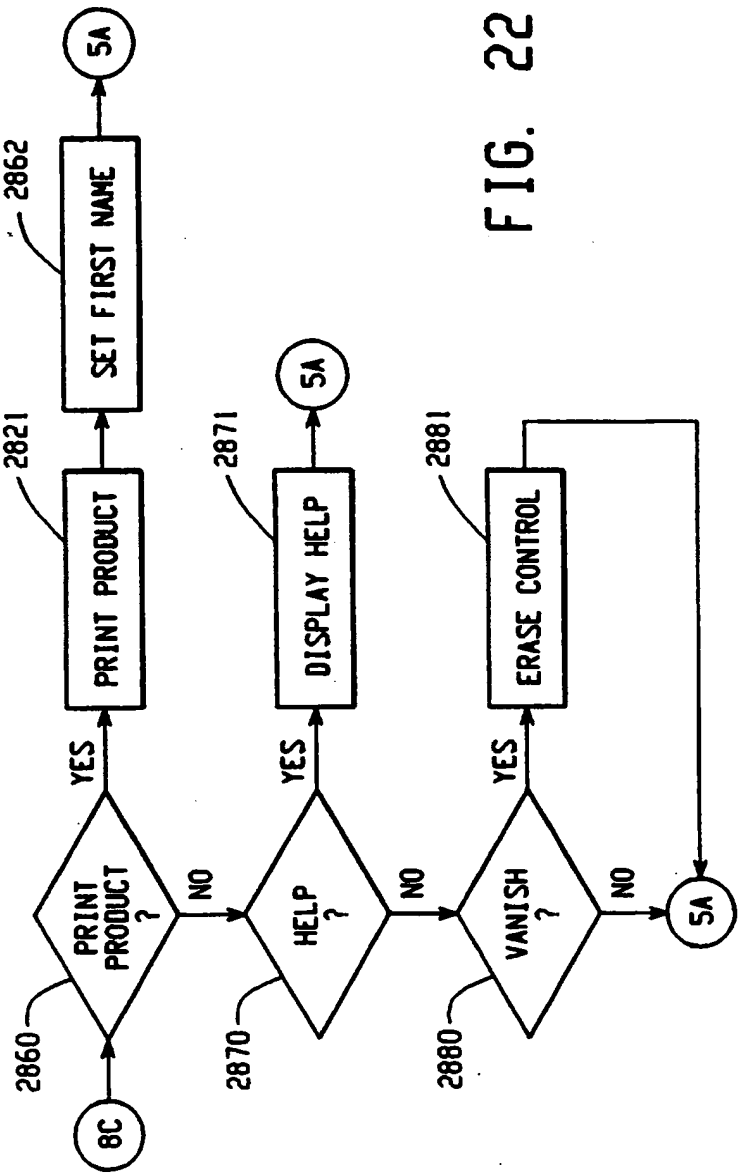

8C
PRINT PRODUCT ?
2860
YES
NO
PRINT PRODUCT
2821
SET FIRST NAME
2862
5A
HELP ?
2870
YES
NO
DISPLAY HELP
2871
5A
VANISH ?
2880
YES
NO
ERASE CONTROL
2881
5A

FIG. 22

INTERNATIONAL SEARCH REPORT

In' tional Application No
PCT/US 95/10518

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/60 G07F17/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO,A,93 16443 (ADVANCED PROMOTION TECHNOLOGIE) 19 August 1993	17-19
A	see claims 1,2,10,15,16 ---	1,2
X	FR,A,2 682 502 (EMQUAD INTERNATIONAL DIVISION FRANCE) 16 April 1993	17-20
A	see page 4, line 1 - line 35 see page 6, line 13 - line 25 ---	1,2
A	EP,A,0 564 736 (HALLMARK CARDS) 13 October 1993 see page 13, line 24 - line 41; figure 15 ---	1,2, 17-19
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 December 1995	03.01.96

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax (+ 31-70) 340-3016	Suendermann, R

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 95/10518

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PATENT ABSTRACTS OF JAPAN vol. 018 no. 441 (P-1788) ,17 August 1994 & JP,A,06 139265 (TOSHIBA CORP) 20 May 1994, see abstract -----	1,2

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

I ational Application No
PCT/US 95/10518

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
WO-A-9316443	19-08-93	AU-B-	3669693	03-09-93
FR-A-2682502	16-04-93	NONE		
EP-A-0564736	13-10-93	AU-B-	1939792	28-10-93
		JP-A-	7192166	28-07-95
		NZ-A-	243390	26-10-95